REMARKS

Enclosed with this response are the following documents required by the Examiner:

1) R. P. Paul, "Robot Manipulators: Mathematics, Programming, and Control" MIT Press Cambridge, MA 1981.
2) Orin and Shrader [1984] "Efficient Computation of the Jacobian for Robot Manipulators".

Applicant's attorney apologizes for the inadvertent omission of these documents in Applicant's response to Paper No. 5.

In view of the above amendments and remarks, this application is believed to be in condition for allowance, which is herewith respectfully requested.

Respectfully submitted,

Kathryn A. Marra, Attorney
Reg. No. 39106
313-665-4708

KAM:ekm
Enclosures

David E. Orin
William W. Schrader
Department of Electrical Engineering
The Ohio State University
Columbus, Ohio 43210

Efficient Computation of the Jacobian for Robot Manipulators

Abstract

This paper discusses and compares six different methods for calculating the Jacobian for a general N-degree-of-freedom manipulator. We enumerate the computational efficiency of each in terms of the total number of multiplications, additions/subtractions, and trigonometric functions required as well as in terms of the number of matrix-vector operations needed. We also give the execution times on a PDP-11/70 minicomputer for determining the Jacobian for an example seven-degree-of-freedom manipulator. This paper formulates one of the best new methods for determining the Jacobian.

1. Introduction

The control for most state-of-the-art industrial robots is relatively simple, based on servos at each joint. However, the controls for the next generation of robots must allow adaptation to the environment through the use of various kinds of sensory information (Dodd and Rossol 1979). For this case, control is most naturally implemented in Cartesian coordinates at the end-effector.

Some kind of transformation is needed between end effector coordinates and joint coordinates. An inverse kinematics procedure may be used to compute joint angles when the end-effector position and orientation are known. However, this may result in extremely complex equations that are difficult to derive and to implement in real time for control (Duffy 1980). Furthermore, for the case of a redundant manipulator (more than 6 degrees of freedom), the inverse kinematics equations are in general underspecified and other approaches to the control based on the Jacobian have been proposed (Liegeois 1977; Klein and Huang 1983; Trevelyan, Kovesi, and Ong 1983; Yoshikawa 1983).

The Jacobian relates joint rates to end-effector rates and may be used if control is based on resolved rate (Whitney 1969) and if position feedback is closed at the end-effector (Klein and Briggs 1980). Also, the transpose of the Jacobian relates end-effector forces to joint torques and may be used when force control is needed (Wu and Paul 1982).

This paper compares several approaches that have been proposed for computing the Jacobian, including a new approach introduced here. These approaches are compared in terms of their computational efficiency since the Jacobian must be computed in real time for control. An enumeration is made of the total number of multiplications, additions/subtractions, and trigonometric functions that are required to determine the Jacobian. These are given as a function of the total number of degrees of freedom, N. The best approaches give computation that varies linearly with N.

The execution times for determining the Jacobian (using PASCAL on a PDP-11/70 minicomputer) for an example seven-degree-of-freedom manipulator is also presented. Finally, the number of matrix-vector operations involved in computing the Jacobian for each approach is enumerated. These may provide the most important basis of comparison for future matrix-vector processors.

1.1. Notation

The basic parameters used to describe the kinematics of a manipulator, as first presented by Denavit and Hartenberg (1955) and extended and given in detail in Paul (1981) are used throughout. In particular, a coordinate system is attached to each link of the manipulator with the z-axis directed along the joint axis. The links are numbered from 0 at the base to N at the end-effector. A separate coordinate system, labeled with the subscript $N + 1$ (or E for end-effector) is also fixed to the end-effector at any desired point.

The International Journal of Robotics Research,
Vol. 3, No. 4, Winter 1984,
0278-3649/84/040066-10 $05.00/0.

Fig. 1. Link parameters associated with link i.

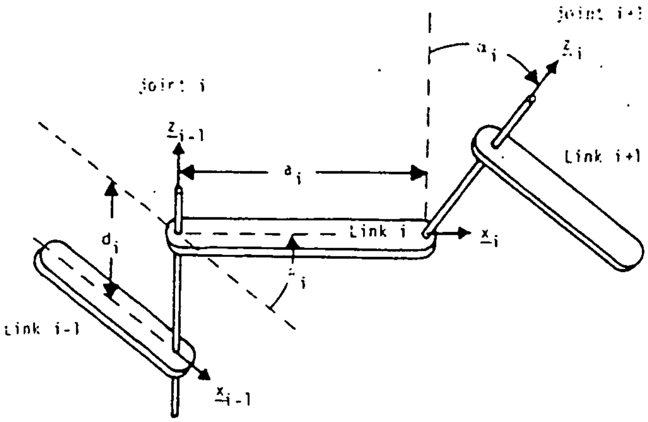
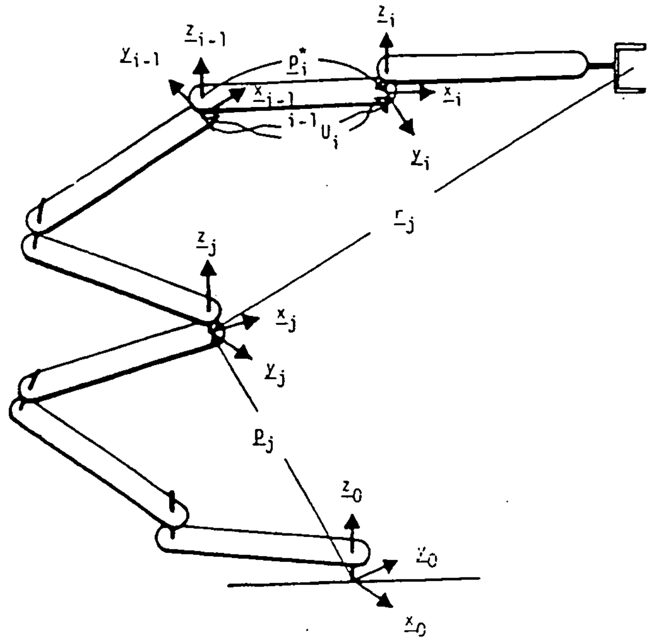


Fig. 2. Depiction of vectors p_j, r_j, p_i^*, and transform ${}^{i-1}U_i$ for a manipulator.



The relative position of joint i, q_i, is with respect to the z-axis of the previous link, $\mathbf{z}_{i-1}$. That is, links $i-1$ and i are connected at joint i.

Four parameters are used to describe each successive joint and link pair—the joint angle (θ_i) and offset distance (d_i) as well as the link length (a_i) and twist (α_i) (see Fig. 1). From these parameters, the 4×4 homogeneous transformation, ${}^{i-1}\mathbf{T}_i$, which relates positions in coordinate system i to those in coordinate system $i-1$, may be computed. For either a rotational or a sliding joint the result is as follows (Paul 1981):

$$
{}^{i-1}\mathbf{T}_i = \left[\begin{array}{ccc|c} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ \hline 0 & 0 & 0 & 1 \end{array}\right], \qquad (1)
$$

where $c\theta_i$ and $s\theta_i$ indicate the cosine and sine, respectively. The transformation between other link coordinate systems may be obtained through multiplication of the intermediate transformation matrices.

The ${}^{i-1}\mathbf{T}_i$ matrix gives both position and orientation changes between successive coordinate systems. The top-left 3×3 part of the matrix gives all orientation information and will be denoted as ${}^{i-1}\mathbf{U}_i$. The first three elements of the right-hand column give all the relative position information and will be denoted as ${}^{i-1}\mathbf{p}_i^*$. Also of value in describing the various approaches to computing the Jacobian are the use of position vectors from the end-effector to link j, $\mathbf{r}_j$, and from the base to link j, $\mathbf{p}_j$. These are all shown in Fig. 2.

2. The Jacobian Matrix

The final form of the Jacobian matrix, as determined in the various approaches, may differ in two important ways:

1. The components may be expressed in any coordinate system from 0 to $N+1$.
2. The reference point on the end-effector (either real or ficticious) for which the translational velocity is instantaneously computed may be chosen somewhat arbitrarily.

These lead to the following notation for the Jacobian matrix, $\mathbf{J}$, and the variables it relates:

$$
\begin{bmatrix} {}^{\ell}\omega \\ {}^{\ell}\mathbf{v} \end{bmatrix}_m = {}^{\ell}\mathbf{J}_m[\dot{\mathbf{q}}], \qquad 0 \le \ell, m \le N+1, \qquad (2)
$$

where

$$
{}^{\ell}\mathbf{J}_m = \begin{bmatrix} {}^{\ell}\gamma_1^x & {}^{\ell}\gamma_2^x & \cdot & \cdot & \cdot & {}^{\ell}\gamma_N^x \\ {}^{\ell}\gamma_1^y & {}^{\ell}\gamma_2^y & \cdot & & & {}^{\ell}\gamma_N^y \\ {}^{\ell}\gamma_1^z & \cdot & & \cdot & & {}^{\ell}\gamma_N^z \\ {}^{\ell}\beta_1^x & \cdot & & & \cdot & {}^{\ell}\beta_N^x \\ {}^{\ell}\beta_1^y & \cdot & & & \cdot & {}^{\ell}\beta_N^y \\ {}^{\ell}\beta_1^z & & & & & {}^{\ell}\beta_N^z \end{bmatrix}_m \quad (3)
$$

and ω is the rotational velocity vector of the end-effector, $\mathbf{v}$ is the translational velocity vector of the end-effector, and $\dot{\mathbf{q}}$ is the vector of joint rates. The components of the end-effector velocity and the Jacobian are expressed in the ℓth link coordinate system, as indicated by the leading superscript. The reference point for the end-effector velocity is understood to be instantaneously at the origin of coordinate system m, as indicated by the trailing subscript. For m equal to other than N or $N + 1$, the velocity reference point is a fictitious point not physically on the end-effector. However, it may be chosen in this way to reduce the computational complexity of determining the Jacobian.

In the following sections, six different methods for computing the Jacobian matrix are presented. In all cases, the fundamental approach from the appropriate reference is given. The notation, however, is standardized, and nonessential details (for purposes of this comparison) are eliminated.

Several of the methods for computing the Jacobian are based on earlier work by Pieper (1968) and Whitney (1972). In fact, in many cases the latest work is just a detailed elaboration of the previous work. However, since the exact computational approach is not made explicit in Pieper and since closer attention to computational detail has resulted in somewhat greater efficiency than that presented in Whitney, these earlier methods will not be compared in this paper.

2.1. Method I

In the first method, as presented by Vukobratović and Potkonjak (1979), the angular velocity of link i is written as a linear function of the previous joint velocities:

$$
{}^{i}\omega_i = \sum_{j=1}^{i} {}^{i}\gamma_j \dot{q}_j . \quad (4)
$$

For a rotational joint, the angular velocity of link i may be computed from that for link $i - 1$ and the relative rate at joint i:

$$
\omega_i = \omega_{i-1} + \dot{q}_i \mathbf{z}_{i-1} . \quad (5)
$$

The appropriate component equation is

$$
{}^{i}\omega_i = {}^{i-1}\mathbf{U}_i^T \left\{ {}^{i-1}\omega_{i-1} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \dot{q}_i \right\} . \quad (6)
$$

The general form for ${}^{i-1}\omega_{i-1}$,

$$
{}^{i-1}\omega_{i-1} = \sum_{j=1}^{i-1} {}^{i-1}\gamma_j \dot{q}_j , \quad (7)
$$

may be used in Eq. (6), with the following result:

$$
{}^{i}\omega_i = \sum_{j=1}^{i-1} [{}^{i-1}\mathbf{U}_i^T \, {}^{i-1}\gamma_j \dot{q}_j] + {}^{i-1}\mathbf{U}_i^T \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \dot{q}_i . \quad (8)
$$

Equating the coefficients for Eqs. (4) and (8) results in a recursive method for computing ${}^{i}\gamma_j$. Using a similar approach for the translational velocity (with coefficients ${}^{i}\beta_j$) and considering the case of sliding joints result in the following sets of recursive equations:

$$
{}^{i}\gamma_i = {}^{i-1}\mathbf{U}_i^T \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} , \quad (9)
$$

$i = 1, 2, \ldots, N)$ revolute joint;

$$
{}^{i}\gamma_i = 0, \quad (10)
$$

$i = 1, 2, \ldots, N)$ sliding joint;

$$
{}^{i}\gamma_j = {}^{i-1}\mathbf{U}_i^T \, {}^{i-1}\gamma_j , \quad (11)
$$

$j = 1, 2, \ldots, i - 1,$
$i = 2, \ldots, N + 1;$

$$
{}^{i}\beta_i = {}^{i}\gamma_i \times {}^{i}\mathbf{p}_i^* , \quad (12)
$$

$i = 1, 2, \ldots, N)$ revolute joint;

$$
{}^{i}\beta_i = {}^{i-1}\mathbf{U}_i^T \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad i = 1, 2, \ldots, N\} \quad \text{sliding joint;} \tag{13}
$$

$$
{}^{i}\beta_j = {}^{i-1}\mathbf{U}_i^T\, {}^{i-1}\beta_j + ({}^{i}\gamma_j \times {}^{i}\mathbf{p}_i^*), \quad j = 1, 2, \ldots, i-1, \quad i = 2, \ldots, N+1. \tag{14}
$$

For $i = N + 1$ in Eqs. (11) and (14), the coefficients as determined are just the elements of the Jacobian matrix, ${}^{N+1}\mathbf{J}_{N+1} = {}^{E}\mathbf{J}_E$. This method is rather inefficient because for $i = N + 1$, $N + 1$ Jacobians are computed. First, the Jacobian is determined for a manipulator consisting of the first link only, then for a manipulator consisting of the first two links, then three links, and so forth up to N links plus the end-effector. The recursions in all cases are from the base to the end-effector. This is a rather inefficient method because all that is needed is the Jacobian for the entire manipulator. It may be very useful, however, in the study of dynamics, where it is necessary to compute the Jacobian for each set of links in the manipulator (Vukobratović and Potkonjak 1979).

2.2. Methods II, III, and IV

The next three methods use the same basic concepts but result in different forms for the Jacobian matrix. They are based on earlier methods by Pieper (1968) and Whitney (1972). With close attention to computational detail, they provide relatively efficient methods for computing the Jacobian.

The angular velocity for the end-effector may be written as a function of the joint rates (for rotational joints):

$$
\omega = \dot{q}_1\mathbf{z}_0 + \dot{q}_2\mathbf{z}_1 + \cdots + \dot{q}_N\mathbf{z}_{N-1}. \tag{15}
$$

The translational velocity for the end-effector may be determined by noting that a particular component is just the appropriate cross-product of the individual joint rate vector and position vector from the joint axis to the velocity reference point ($-\mathbf{t}_i$). That is,

$$
\mathbf{v} = -(\dot{q}_1\mathbf{z}_0) \times \mathbf{t}_0 - (\dot{q}_2\mathbf{z}_1) \times \mathbf{t}_1 - \cdots - (\dot{q}_N\mathbf{z}_{N-1}) \times \mathbf{t}_{N-1}. \tag{16}
$$

Comparing Eqs. (2) and (3) with Eqs. (15) and (16), we find that the coefficients of the Jacobian matrix, for the case of a rotational joint, may be written in terms of the joint axis vector and relative position vector:

$$
\left.\begin{aligned} \gamma_i &= \mathbf{z}_{i-1} \\ \beta_i &= -(\mathbf{z}_{i-1} \times \mathbf{t}_{i-1}) \end{aligned}\right\} \text{rotational joint.} \tag{17}
$$

For the case of a sliding joint, similar expressions may be derived:

$$
\left.\begin{aligned} \gamma_i &= 0, \\ \beta_i &= \mathbf{z}_{i-1} \end{aligned}\right\} \text{sliding joint.} \tag{18}
$$

The three methods presented in this section differ in the coordinate system used to express the components of the Jacobian and also in the velocity reference point chosen. In Waldron's work (1982), the components of the Jacobian are determined with respect to the base coordinate system 0, and the velocity reference point is chosen to be the origin of the base coordinate system. ${}^{0}\mathbf{J}_0$ is thus computed. Since $\mathbf{p}_i$ is defined as the position vector from the origin of the base coordinate system to that of link i (see Fig. 2), then the following recursive equations may be used:

$$
{}^{0}\mathbf{U}_0 = \mathbf{I}; \tag{19}
$$

$$
{}^{0}\mathbf{U}_i = {}^{0}\mathbf{U}_{i-1}\, {}^{i-1}\mathbf{U}_i, \quad i = 1, 2, \ldots, N-1; \tag{20}
$$

$$
{}^{0}\gamma_i = {}^{0}\mathbf{U}_{i-1} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad i = 1, 2, \ldots, N\} \quad \text{revolute joint;} \tag{21}
$$

$$
{}^{0}\gamma_i = \mathbf{0}, \quad i = 1, 2, \ldots, N\} \quad \text{sliding joint;} \tag{22}
$$

$$
{}^{0}\mathbf{p}_0 = \mathbf{0}; \tag{23}
$$

$$
{}^0\mathbf{p}_i = {}^0\mathbf{p}_{i-1} + {}^0\mathbf{U}_i\,{}^i\mathbf{p}_i^*, \qquad i = 1, 2, \ldots, N-1; \tag{24}
$$

$$
{}^0\boldsymbol{\beta}_i = {}^0\boldsymbol{\gamma}_i \times (-{}^0\mathbf{p}_{i-1}), \qquad i = 1, 2, \ldots, N\} \quad \text{revolute joint;} \tag{25}
$$

$$
{}^0\boldsymbol{\beta}_i = {}^0\mathbf{U}_{i-1}\begin{bmatrix}0\\0\\1\end{bmatrix}, \qquad i = 1, 2, \ldots, N\} \quad \text{sliding joint.} \tag{26}
$$

Note in the above equations that the components of $\mathbf{z}_{i-1}$ in coordinate system $i-1$ are $[0 \quad 0 \quad 1]^T$. Also, $\mathbf{I}$ is the identity matrix. In Eqs. (20) and (24), the recursions for computing the orientation matrix and position vector are directed from the base out to the end-effector.

In Olson and Ribble's work (Ribble 1982), the components of the Jacobian are determined with respect to the base coordinate system, and the velocity reference point is chosen to be the center of the end-effector E. ${}^0\mathbf{J}_E$ is thus computed. Since $\mathbf{r}_i$ is defined as the position vector from the origin of the end-effector coordinate system to that of link i, Eq. (25) may be replaced by

$$
\left.\begin{aligned}
{}^0\mathbf{r}_{i-1} &= {}^0\mathbf{p}_{i-1} - {}^0\mathbf{p}_{N+1} \qquad (27)\\
{}^0\boldsymbol{\beta}_i &= {}^0\boldsymbol{\gamma}_i \times (-{}^0\mathbf{r}_{i-1}) \qquad (28)
\end{aligned}\right\} \quad i = 1, \ldots, N.
$$

Also, the indices on Eqs. (20) and (24) must be changed to $N+1$. In Olson and Ribble's work the recursion for computing the orientation matrix is from the base out. The recursion for the position vector is also from the base out; however, Eq. (27) must be added to give the velocity reference point at the center of the end-effector. While this method is slower than that of Waldron (1982), an advantage is that the velocity reference point is a real point on the end-effector and so the values of ${}^0\boldsymbol{\beta}_i$ relate directly to the velocity of the end-effector.

In Renaud's work (1981), the components of the Jacobian and the velocity reference point are both associated with a link ℓ, which is approximately midway between the base and end-effector. ${}^\ell\mathbf{J}_\ell$ is thus computed. The recursive equations for computing γ and β are

$$
{}^\ell\mathbf{U}_\ell = \mathbf{I}; \tag{29}
$$

$$
{}^\ell\mathbf{U}_i = {}^\ell\mathbf{U}_{i-1}\,{}^{i-1}\mathbf{U}_i, \qquad i = \ell+1, \ell+2, \ldots, N-1; \tag{30}
$$

$$
{}^\ell\mathbf{U}_{i-1} = {}^\ell\mathbf{U}_i\,{}^{i-1}\mathbf{U}_i^T, \qquad i = \ell, (\ell-1), \ldots, 1; \tag{31}
$$

$$
{}^\ell\boldsymbol{\gamma}_i = {}^\ell\mathbf{U}_{i-1}\begin{bmatrix}0\\0\\1\end{bmatrix}, \qquad i = 1, 2, \ldots, N\} \quad \text{revolute joint;} \tag{32}
$$

$$
{}^\ell\boldsymbol{\gamma}_i = \mathbf{0}, \qquad i = 1, 2, \ldots, N\} \quad \text{sliding joint;} \tag{33}
$$

$$
{}^\ell\mathbf{t}_\ell = \mathbf{0}; \tag{34}
$$

$$
{}^\ell\mathbf{t}_i = {}^\ell\mathbf{t}_{i-1} + {}^\ell\mathbf{U}_i\,{}^i\mathbf{p}_i^*, \qquad i = \ell+1, \ell+2, \ldots, N-1; \tag{35}
$$

$$
{}^\ell\mathbf{t}_{i-1} = {}^\ell\mathbf{t}_i - {}^\ell\mathbf{U}_i\,{}^i\mathbf{p}_i^*, \qquad i = \ell, (\ell-1), \ldots, 1; \tag{36}
$$

$$
{}^\ell\boldsymbol{\beta}_i = {}^\ell\boldsymbol{\gamma}_i \times (-{}^\ell\mathbf{t}_{i-1}), \qquad i = 1, 2, \ldots, N\} \quad \text{revolute joint;} \tag{37}
$$

$$
{}^\ell\boldsymbol{\beta}_i = {}^\ell\mathbf{U}_{i-1}\begin{bmatrix}0\\0\\1\end{bmatrix}, \qquad i = 1, 2, \ldots, N\} \quad \text{sliding joint.} \tag{38}
$$

The position vector $\mathbf{t}_i$ in the previous equations is from the origin of coordinate system $m = \ell$ to the origin of coordinate system i. Also note that the recursions are from link ℓ in toward the base and out toward the end-effector.

The main advantage of this method is that the manipulator is broken into two manipulators of approximate length $N/2$, and the Jacobian for a manipulator of length $N/2$ is calculated twice. The matrices at the beginning of the Jacobian calculation are simple, and Renaud's method, starting twice, takes advantage of this.

2.3. Method V

The method described in this section is new, resulting from an effort to compute ${}^E\mathbf{J}_E$ in the most efficient

way possible. We see from Eqs. (9) through (14) that Method I is somewhat inefficient because the elements of N different Jacobians are computed. That is, the Jacobian for a manipulator consisting of the first link only is determined, then for a manipulator consisting of the first two links, then the first three links, and so on. Also, the recursions are from the base to the end-effector.

If the recursion is redirected to be from the end-effector to the base and if values of γ and β not needed are not computed, then the resulting equations are

$$^{N+1}\mathbf{U}_{N+1} = \mathbf{I}; \tag{39}$$

$$^{N+1}\mathbf{U}_{i-1} = {}^{N+1}\mathbf{U}_i\, {}^{i-1}\mathbf{U}_i^T, \quad i = (N+1), \ldots, 2, 1; \tag{40}$$

$$^{N+1}\gamma_i = {}^{N+1}\mathbf{U}_{i-1} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad i = 1, 2, \ldots, N\} \text{ revolute joint;} \tag{41}$$

$$^{N+1}\gamma_i = 0, \quad i = 1, 2, \ldots, N\} \text{ sliding joint;} \tag{42}$$

$$^{N+1}\mathbf{r}_{N+1} = 0; \tag{43}$$

$$^{N+1}\mathbf{r}_{i-1} = {}^{N+1}\mathbf{r}_i - {}^{N+1}\mathbf{U}_i\, {}^i\mathbf{p}_i^*, \quad i = (N+1), \ldots, 2, 1; \tag{44}$$

$$^{N+1}\beta_i = {}^{N+1}\gamma_i \times (-{}^{N+1}\mathbf{r}_{i-1}), \quad i = 1, 2, \ldots, N\} \text{ revolute joint;} \tag{45}$$

$$^{N+1}\beta_i = {}^{N+1}\mathbf{U}_{i-1} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad i = 1, 2, \ldots, N\} \text{ sliding joint.} \tag{46}$$

It should be noted that the resulting equations are similar to Waldron's (1982) except that the recursion is in the opposite direction. This was not immediately apparent during the development of the equations. Once the common notation had been applied to all the methods, we recognized the similarity. The velocity reference point is now a real point on the end-effector, and the components of the Jacobian are expressed in the end-effector coordinate system.

2.4. Method VI

The final method comes from Paul's work (1981) and also computes $^{E}\mathbf{J}_E$. Paul, however, uses the 4 × 4 homogeneous transformation matrix as the basis for the computations. The appropriate equations are

$$^{N+1}\mathbf{T}_{N+1} = \mathbf{I}; \tag{47}$$

$$^{i-1}\mathbf{T}_{N+1} = {}^{i-1}\mathbf{T}_i\, {}^i\mathbf{T}_{N+1}, \quad i = (N+1), \ldots, 2, 1, \tag{48}$$

$$^{N+1}\gamma_i = {}^{i-1}\mathbf{U}_{N+1}^T \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad i = 1, 2, \ldots, N\} \text{ revolute joint;} \tag{49}$$

$$^{N+1}\gamma_i = 0, \quad i = 1, 2, \ldots, N\} \text{ sliding joint;} \tag{50}$$

$$^{N+1}\beta_i^j = -[{}^{i-1}\mathbf{U}_{N+1}^j \times {}^{i-1}\mathbf{r}_{i-1}]^T \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad \left.\begin{matrix} j = 1, 2, 3, \\ i = 1, 2, \ldots, N \end{matrix}\right\} \text{ revolute joint;} \tag{51}$$

$$^{N+1}\beta_i = {}^{i-1}\mathbf{U}_{N+1}^T \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad i = 1, 2, \ldots, N\} \text{ sliding joint.} \tag{52}$$

Note that $^{i-1}\mathbf{U}_{N+1}^j$ is just column j of the 3 × 3 orientation part of $^{i-1}\mathbf{T}_{N+1}$. Also note that $^{i-1}\mathbf{r}_{i-1}$ is just the first three elements of the fourth column of $^{i-1}\mathbf{T}_{N+1}$. Furthermore, $^{N+1}\beta_i^j$ is just the jth component of the vector $^{N+1}\beta_i$.

It may be noted that the results for Methods V and VI are quite similar except that in Method VI, the 3 × 3 orientation matrix and 3 × 1 position vector of Method V are grouped into the 4 × 4 homogeneous transformation matrix form. If the bottom row of the homogeneous transformation matrix is understood to be composed of three 0's and a 1, this seems to make little if any difference in the overall computational complexity. However, another difference is that the ordering of matrix multiplication in Eq. (48) is reversed from that of Eq. (40), and this does result in

Table 1. Number of Computations for Each Method Maintaining the Natural Jacobian Form (N = Number of degrees of freedom)

Method	J	Multiplication	Addition/ Subtraction	Sines/ Cosines	N = 6 M	A
I Vukobratović/Potkonjak	$^{E}J_{E}$	$10N^2 - 15N + 9$	$N^2 + 7N - 2$	$2N$	[illegible]	76
II Waldron	$^{0}J_{0}$	$30N - 55$	$15N - 38$	$2N - 2$	[illegible]	52
III Olson/Ribble	$^{0}J_{E}$	$30N - 11$	$18N - 20$	$2N$	169	88
IV Renaud	$^{\ell}J_{\ell}$	$30N - 87$	$15N - 66$	$2N - 2$	93	24
V Orin/Schrader	$^{E}J_{E}$	$30N - 25$	$15N - 25$	$2N$	155	[illegible]
VI Paul	$^{E}J_{E}$	$30N - 18$	$14N - 15$	$2N$	162	69
Wampler	$^{0}J_{E}$	$26N - 28$	$13N - 13$	$2N$	128	65
	$^{E}J_{E}$	$26N - 20$	$13N - 10$	$2N$	136	68

increased complexity in Eq. (51) as opposed to that of Eq. (45). It appears then that some computational savings may be made in Paul's method if the order of matrix multiplication in Eq. (48) is changed.

3. Comparison of the Methods

We now assess the numerical efficiency of computing the Jacobian according to each of the six methods just described. In particular, we enumerate the total number of additions/subtractions, multiplications, and trigonometric functions performed in the process of evaluating the Jacobian. The results, given as a function of the total number of degrees of freedom N, are shown in Table 1.

The results in Table 1 were generated through the use of a program that considered the components of matrices and vectors to be composed entirely of four symbols. The symbol x indicated a nonzero value other than ± 1; + indicated a value of $+1$, − indicated a value of -1, and 0 indicated a value of zero. The program produced a symbolic Jacobian while counting the number of multiplications, additions/subtractions, and trigonometric functions used to produce the Jacobian. Additions, subtractions, and multiplications that included zero were not counted. Multiplications that included ± 1 were totaled separately but have been added into the multiplication formulas of Table 1. It has also been assumed that the twist angle of each link is $0°$ or $\pm 90°$ since most manipulators have this type of kinematic configuration.

From Table 1, we see that the best method is that of Renaud (1981). It should be understood, however,

Table 2. Execution Time for Each Method on the PDP-11/70

Method	J	Execution Time (ms)
I Vukobratović/Potkonjak	$^{E}J_{E}$	23.0
II Waldron	$^{0}J_{0}$	12.8
III Olson/Ribble	$^{0}J_{E}$	16.2
IV Renaud	$^{\ell}J_{\ell}$	12.9
V Orin/Schrader	$^{E}J_{E}$	15.5
VI Paul	$^{E}J_{E}$	16.9

that to use this form of the Jacobian for resolved rate control it is necessary to associate the angular and translational velocity of the end-effector with coordinate system ℓ, where $\ell = (N + 1)/2$.

Methods I, V, and VI all result in the same form for the Jacobian, $^{E}J_{E}$. Of these, the new method proposed in this paper requires the least amount of computation.

All of the methods discussed in this paper have been used to simulate a seven-degree-of-freedom manipulator that has a somewhat general kinematic configuration (nonzero joint offsets and link lengths). A PDP-11/70 minicomputer with floating-point hardware and the PASCAL language were used, and the results are given in Table 2. The results show a close correlation with those of Table 1.

In Table 2, any changes in ordering from that anticipated by reviewing Table 1 probably result from differences in the amount of overhead involved in indexed addressing. Also, in evaluating the execution times given in Table 2, it should be understood that

Table 3. Number of Computations for Each Method Including Conversion to ${}^{E}\mathbf{J}_E$
(N = Number of degrees of freedom)

Method	Multiplication	Addition/ Subtraction	Sines/ Cosines
I Vukobratović/Potkonjak	$10N^2 - 15N + 9$	$N^2 + 7N - 2$	$2N$
II Waldron	$54N - 31$	$33N - 37$	$2N$
III Olson/Ribble	$48N - 23$	$30N - 32$	$2N$
IV Renaud	$54N - 81$	$30N - 79$	$2N$
V Orin/Schrader	$30N - 25$	$15N - 25$	$2N$
VI Paul	$30N - 18$	$14N - 15$	$2N$
Wampler	26N - 20	13N - 10	2N

Table 4. Execution Time for Each Method on the PDP-11/70
(All Forms Further Converted to ${}^{E}\mathbf{J}_E$).

Method	Execution Time (ms)
I Vukobratović/Potkonjak	23.0
II Waldron	20.1
III Olson/Ribble	19.1
IV Renaud	20.1
V Orin/Schrader	15.5
VI Paul	16.9

the absolute magnitudes have little significance. That is, little effort was made to reduce these times. Instead, much effort was made to ensure "fairness" so that relative magnitudes are significant.

Perhaps a more appropriate way to compare the various methods is to transform the Jacobian produced by each method to a common form. In this case, ${}^{E}\mathbf{J}_E$ will be used as the final form. To move the velocity reference point from the origin of coordinate system m to the end-effector, the following transformation is appropriate:

$$
{}^{\ell}\mathbf{J}_E = \left[\begin{array}{ccc|c} & \mathbf{I} & & \mathbf{0} \\ \hline 0 & -{}^{\ell}r_m^z & {}^{\ell}r_m^y & \\ {}^{\ell}r_m^z & 0 & -{}^{\ell}r_m^x & \mathbf{I} \\ -{}^{\ell}r_m^y & {}^{\ell}r_m^x & 0 & \end{array}\right] {}^{\ell}\mathbf{J}_m. \quad (53)
$$

To convert the components of ${}^{\ell}\gamma_{iE}$ and ${}^{\ell}\beta_{iE}$ from the ℓth coordinate system to the end-effector coordinate system, the following transformation may be used:

$$
{}^{E}\mathbf{J}_E = \left[\begin{array}{c|c} {}^{\ell}\mathbf{U}_E^T & 0 \\ \hline 0 & {}^{\ell}\mathbf{U}_E^T \end{array}\right] {}^{\ell}\mathbf{J}_E. \quad (54)
$$

The total transformation is just a projective transformation of screw coordinates (Sugimoto and Matsumoto 1983) and is given as follows:

$$
{}^{E}\mathbf{J}_E = \left[\begin{array}{c|c} {}^{\ell}\mathbf{U}_E^T & 0 \\ \hline {}^{\ell}\mathbf{U}_E^T * \begin{bmatrix} 0 & -{}^{\ell}r_m^z & {}^{\ell}r_m^y \\ {}^{\ell}r_m^z & 0 & -{}^{\ell}r_m^x \\ -{}^{\ell}r_m^y & {}^{\ell}r_m^x & 0 \end{bmatrix} & {}^{\ell}\mathbf{U}_E^T \end{array}\right] {}^{\ell}\mathbf{J}_m. \quad (55)
$$

The results for each method after conversion to ${}^{E}\mathbf{J}_E$ are given in Tables 3 and 4. From these tables, we see that the last two methods are the most efficient, with the one proposed in this paper being the best.

It should also be understood that the results presented in these tables are not completely definitive. Each method has its own natural output form for the Jacobian, and to depart from this to transform it to a common form biases the results toward those that naturally produce this common form. However, whether the natural or common form of the Jacobian is used, Methods V and VI are especially computationally efficient.

Table 5. Number of Matrix-Vector Computations for Each Method

Method	J	Cross-Product	Matrix-Vector Product	Matrix Product (3×3)	Vector Addition/ Subtraction	Matrix Product (4×4)
I Vukobratović/Potkonjak	${}^{E}\mathbf{J}_{E}$	$1/2(N^2 + 3N)$	$N^2 + N$		$1/2(N^2 + N)$	
II Waldron	${}^{0}\mathbf{J}_{0}$	$N - 1$	$N - 1$	$N - 2$	$N - 2$	
III Olson/Ribble	${}^{0}\mathbf{J}_{E}$	N	$N + 1$	N	$2N$	
IV Renaud	${}^{t}\mathbf{J}_{t}$	$N - 1$	$N - 2$	$N - 3$	$N - 3$	
V Orin/Schrader	${}^{E}\mathbf{J}_{E}$	N	N	N	N	
VI Paul	${}^{E}\mathbf{J}_{E}$	$3N$				N

One final comparison among the methods is given in Table 5. In this case, the basic arithmetic operations are for vectors and matrices. This is appropriate because processors of the future should be equipped with such instructions. The results are similar to those obtained previously, in that Renaud's method is the best. Of those that result in ${}^{E}\mathbf{J}_{E}$, the method proposed in this paper is the best. (A 4×4 matrix product of homogeneous transforms is basically equivalent to a 3×3 matrix product along with a matrix-vector product and vector add.)

4. Summary

Six different methods for computing the Jacobian have been presented and compared for computational efficiency. The results indicate that the new method presented in this paper is the most computationally efficient when the Jacobian is based on end-effector coordinates. A method proposed by Renaud (1981) gives the best results when an arbitrary coordinate system is used.

To use a Jacobian of any particular form for resolved rate control or force control, the end-effector rates or forces must be associated with the same coordinate system. The total amount of computation required often depends on the specifics of the application for which the Jacobian is used. At a minimum, though, the results of this paper should give a general indication of the total computation required.

The results presented here assume the use of a serial processor. A pipelined linear array of processors may also be used but requires a new formulation of the problem. This is the subject of present investigations at The Ohio State University (Schrader 1983), and the present paper provides the foundation for this work. Advances in both algorithms and architectures promise to allow a control based on the Jacobian to be more viable for real-time implementation in the future.

Acknowledgment

This work was supported by the National Science Foundation under Grant No. ECS-8312677 to The Ohio State University.

References

Denavit, J., and Hartenberg, R. B. 1955. A kinematic notation for lower-pair mechanisms based on matrices. *ASME J. Appl. Mech.*, vol. 23, pp. 215–221.

Dodd, G. G., and Rossol, L., eds. 1979. *Computer vision and sensor-based robots.* New York: Plenum Press.

Duffy, J. 1980. *Analysis of mechanisms and robot manipulators.* London: Edward Arnold.

Klein, C. A., and Briggs, R. L. 1980. Use of active compliance in the control of legged vehicles. *IEEE Trans. Sys. Man Cyber.* SMC-10(7):393–400.

Klein, C. A., and Huang, C. H. 1983. Review of pseudoinverse control for use with kinematically redundant manipulators. *IEEE Trans. Sys. Man Cyber.* SMC-13(3):245–250.

Liegeois, A. 1977. Automatic supervisory control of the configuration and behaviour of multibody mechanisms. *IEEE Trans. Sys. Man Cyber.* SMC-7(12):868–871.

Paul, R. P. 1981. *Robot manipulators: Mathematics, programming, and control.* Cambridge: MIT Press.

Pieper, D. L. 196 . The kinematics of manipulators under computer cont ol. Ph.D. thesis. Stanford University. Department of C mputer Science.

Renaud, M. 198 . (Oct., Tokyo). Geometric and kinematic models of a rol ot manipulator: Calculation of the Jacobian matrix and its inverse. *Proc. 11th Int. Symp. Industr. Robots.*

Ribble, E. A. 1982. Synthesis of human skeletal motion and the design of a special-purpose processor for real-time animation of human and animal figure motion. M.S. thesis, Ohio State University, Department of Electrical Engineering, Columbus, Ohio.

Schrader, W. W. 1983. Efficient Jacobian computation for robot manipulators on serial and pipelined processors. M.S. thesis, Ohio State University, Department of Electrical Engineering, Columbus, Ohio.

Sugimoto, K., and Matsumoto, Y. 1983 (Bretton Woods, N.H.). Kinematic analysis of manipulators by means of the projective transformation of screw coordinates. *Proc. 1st Int. Symp. Robotics Res.* Cambridge: M.I.T. Artificial Intelligence Laboratory.

Trevelyan, Kovesi, and Ong 1983; Yoshikawa 1983). (Bretton Woods, N.H.). Motion control for a sheep shearing robot. *Proc. 1st Int. Symp. Robotics Res.* Cambridge: M.I.T. Artificial Intelligence Laboratory.

Vukobratović, M., and Potkonjak, V. 1979. Contribution of the forming of computer methods for automatic modelling of spatial mechanisms motions. *Mechanism and Machine Theory*, vol. 14, pp. 179–200.

Waldron, K. J. 1982. Geometrically based manipulator rate control algorithms. *Mechanism and Machine Theory* 17(6):379–385.

Whitney, D. E. 1969. Resolved motion rate control of manipulators and human prostheses. *IEEE Trans. Man-Mach. Sys.* MMS-10(2):47–53.

Whitney, D. E. 1972. The mathematics of coordinated control of prosthetic arms and manipulators. *J. Dyn. Sys. Measurement. Contr.*, vol. 94, pp. 303–309.

Wu, C. H., and Paul, R. P. 1982. Resolved motion force control of robot manipulators, *IEEE Trans. Sys. Man Cyber.* SMC-12(3):266–275.

Yoshikawa, T. 1983 (Bretton Woods, N.H.). Analysis and control of robot manipulators with redundancy. *Proc. 1st Int. Symp. Robotics Res.* Cambridge: M.I.T. Artificial Intelligence Laboratory.

The MIT Press Series in Artificial Intelligence

Artificial Intelligence: An MIT Perspective, Volume I: Expert Problem Solving, Natural Language Understanding, Intelligent Computer Coaches, Representation and Learning edited by Patrick Henry Winston and Richard Henry Brown, 1979

Artificial Intelligence: An MIT Perspective, Volume II: Understanding Vision, Manipulation, Computer Design, Symbol Manipulation edited by Patrick Henry Winston and Richard Henry Brown, 1979

NETL: A System for Representing and Using Real-World Knowledge by Scott Fahlman, 1979

The Interpretation of Visual Motion by Shimon Ullman, 1979

A Theory of Syntactic Recognition for Natural Language by Mitchell P. Marcus, 1980

Turtle Geometry: The Computer as a Medium for Exploring Mathematics by Harold Abelson and Andrea diSessa, 1981

From Images to Surfaces: A Computational Study of the Human Early Visual System by William Eric Leifur Grimson, 1981

Robot Manipulators: Mathematics, Programming and Control by Richard P. Paul, 1981

ROBOT MANIPULATORS: MATHEMATICS, PROGRAMMING, AND CONTROL

The Computer Control of Robot Manipulators

Richard P. Paul

The MIT Press
Cambridge, Massachusetts and London, England

PUBLISHER'S NOTE

This format is intended to reduce the cost of publishing certain works in book form and to shorten the gap between editorial preparation and final publication. The time and expense of detailed editing and composition in print have been avoided by photographing the text of this book directly from the author's computer printout.

Printed in the United States of America.

```
Library of Congress Cataloging in Publication Data

Paul, Richard P.
   Robot manipulators.

   (The MIT Press series in artificial intelligence)
   Bibliography: p.
   Includes index.
   1. Robots, Industrial.  2. Manipulators
(Mechanism)  I. Title.  II. Series.
TS191.8.P38         629.8'92           81-13689
ISBN 0-262-16082-X                     AACR2
```


CONTENTS

CHAPTER 1

HOMOGENEOUS TRANSFORMATIONS

1.1 Introduction

The study of robot manipulation is concerned with the relationship between objects, and between objects and manipulators. In this chapter we will develop the representation necessary to describe these relationships. Similar problems of representation have already been solved in the field of computer graphics, where the relationship between objects must also be described. Homogeneous transformations are used in this field and in computer vision [Duda] [Roberts63] [Roberts65]. These transformations were employed by Denavit to describe linkages [Denavit] and are now used to describe manipulators [Pieper] [Paul72] [Paul77b].

We will first establish notation for vectors and planes and then introduce transformations on them. These transformations consist primarily of translation and rotation. We will then show that these transformations can also be considered as coordinate frames in which to represent objects, including the manipulator. The inverse transformation will then be introduced. A later section describes the general rotation transformation representing a rotation about a vector. An algorithm is then described to find the equivalent axis and angle of rotation represented by any given transformation. A brief section on stretching and scaling transforms is included together with a section on the perspective transformation. The chapter concludes with a section on transformation equations.

1.2 Notation

In describing the relationship between objects we will make use of point vectors, planes, and coordinate frames. Point vectors are denoted by lower case, bold face characters. Planes are denoted by script characters, and coordinate frames by upper case, bold face characters. For example:

vectors	**v**, **x1**, **x**
planes	$\mathcal{P}$, $\mathcal{Q}$
coordinate frames	**I**, **A**, **CONV**

We will use point vectors, planes, and coordinate frames as variables which have associated values. For example, a point vector has as value its three

Cartesian coordinate components.

If we wish to describe a point in space, which we will call p, with respect to a coordinate frame E, we will use a vector which we will call v. We will write this as

$$^{E}\mathbf{v}$$

The leading superscript describes the defining coordinate frame.

We might also wish to describe this same point, p, with respect to a different coordinate frame, for example H, using a vector w as

$$^{H}\mathbf{w}$$

v and w are two vectors which probably have different component values and $\mathbf{v} \neq \mathbf{w}$ even though both vectors describe the same point p. The case might also exist of a vector a describing a point 3 inches above any frame

$$^{F1}\mathbf{a} \qquad ^{F2}\mathbf{a}$$

In this case the vectors are identical but describe different points. Frequently, the defining frame will be obvious from the text and the superscripts will be left off. In many cases the name of the vector will be the same as the name of the object described, for example, the tip of a pin might be described by a vector **tip** with respect to a frame BASE as

$$^{BASE}\mathbf{tip}$$

If it were obvious from the text that we were describing the vector with respect to BASE then we might simply write

$$\mathbf{tip}$$

If we also wish to describe this point with respect to another coordinate frame say, HAND, then we must use another vector to describe this relationship, for example

$$^{HAND}\mathbf{tv}$$

$^{HAND}\mathbf{tv}$ and **tip** both describe the same feature but have different values. In order to refer to individual components of coordinate frames, point vectors, or planes, we add subscripts to indicate the particular component. For example, the vector $^{HAND}\mathbf{tv}$ has components $^{HAND}\mathbf{tv}_x$, $^{HAND}\mathbf{tv}_y$, $^{HAND}\mathbf{tv}_z$.

1.3 Vectors

The homogeneous coordinate representation of objects in n-space is an $(n+1)$-space entity such that a particular perspective projection recreates the n-space. This can also be viewed as the addition of an extra coordinate to each vector, a scale factor, such that the vector has the same meaning if each component, including the scale factor, is multiplied by a constant.

A point vector

$$\mathbf{v} = a\mathbf{i} + b\mathbf{j} + c\mathbf{k} \tag{1.1}$$

where i, j, and k are unit vectors along the x, y, and z coordinate axes, respectively, is represented in homogeneous coordinates as a column matrix

$$\mathbf{v} = \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \tag{1.2}$$

where

$$\begin{aligned} a &= x/w \\ b &= y/w \\ c &= z/w \end{aligned} \tag{1.3}$$

Thus the vector $3\mathbf{i} + 4\mathbf{j} + 5\mathbf{k}$ can be represented as $[3, 4, 5, 1]^T$ or as $[6, 8, 10, 2]^T$ or again as $[-30, -40, -50, -10]^T$, etc. The superscript T indicates the transpose of the row vector into a column vector. The vector at the origin, the null vector, is represented as $[0, 0, 0, n]^T$ where n is any non-zero scale factor. The vector $[0, 0, 0, 0]^T$ is undefined. Vectors of the form $[a, b, c, 0]^T$ represent vectors at infinity and are used to represent directions; the addition of any other finite vector does not change their value in any way.

We will also make use of the vector dot and cross products. Given two vectors

$$\begin{aligned} \mathbf{a} &= a_x\mathbf{i} + a_y\mathbf{j} + a_z\mathbf{k} \\ \mathbf{b} &= b_x\mathbf{i} + b_y\mathbf{j} + b_z\mathbf{k} \end{aligned} \tag{1.4}$$

we define the vector dot product, indicated by "·" as

$$\mathbf{a} \cdot \mathbf{b} = a_x b_x + a_y b_y + a_z b_z \tag{1.5}$$

The dot product of two vectors is a scalar. The cross product, indicated by a "×", is another vector perpendicular to the plane formed by the vectors of the product and is defined by

$$\mathbf{a} \times \mathbf{b} = (a_y b_z - a_z b_y)\mathbf{i} + (a_z b_x - a_x b_z)\mathbf{j} + (a_x b_y - a_y b_x)\mathbf{k} \tag{1.6}$$

This definition is easily remembered as the expansion of the determinant

$$\mathbf{a} \times \mathbf{b} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ a_x & a_y & a_z \\ b_x & b_y & b_z \end{vmatrix} \tag{1.7}$$

1.4 Planes

A plane is represented as a row matrix

$$\mathcal{P} = [a, b, c, d] \tag{1.8}$$

such that if a point v lies in a plane $\mathcal{P}$ the matrix product

$$\mathcal{P}\mathrm{v} = 0 \tag{1.9}$$

or in expanded form

$$xa + yb + zc + wd = 0 \tag{1.10}$$

If we define a constant

$$m = +\sqrt{a^2 + b^2 + c^2} \tag{1.11}$$

and divide Equation 1.10 by wm we obtain

$$\frac{x}{w}\frac{a}{m} + \frac{y}{w}\frac{b}{m} + \frac{z}{w}\frac{c}{m} = -\frac{d}{m} \tag{1.12}$$

The left hand side of Equation 1.12 is the vector dot product of two vectors $(x/w)\mathrm{i} + (y/w)\mathrm{j} + (z/w)\mathrm{k}$ and $(a/m)\mathrm{i} + (b/m)\mathrm{j} + (c/m)\mathrm{k}$ and represents the directed distance of the point $(x/w)\mathrm{i} + (y/w)\mathrm{j} + (z/w)\mathrm{k}$ along the vector $(a/m)\mathrm{i} + (b/m)\mathrm{j} + (c/m)\mathrm{k}$. The vector $(a/m)\mathrm{i} + (b/m)\mathrm{j} + (c/m)\mathrm{k}$ can be interpreted as the outward pointing normal of a plane situated a distance $-d/m$ from the origin in the direction of the normal. Thus a plane $\mathcal{P}$ parallel to the x, y plane, one unit along the z axis, is represented as

$$\mathcal{P} = [0, 0, 1, -1] \tag{1.13}$$

or as $$\mathcal{P} = [0, 0, 2, -2] \tag{1.14}$$

or as $$\mathcal{P} = [0, 0, -100, 100] \tag{1.15}$$

A point $\mathrm{v} = [10, 20, 1, 1]$ should lie in this plane

$$[0, 0, -100, 100]\begin{bmatrix}10\\20\\1\\1\end{bmatrix} = 0 \tag{1.16}$$

or

$$\begin{bmatrix}0 & 0 & 1 & -1\end{bmatrix}\begin{bmatrix}-5\\-10\\-.5\\-.5\end{bmatrix} = 0 \tag{1.17}$$

The point $\mathrm{v} = [0, 0, 2, 1]$ lies above the plane

$$[0, 0, 2, -2]\begin{bmatrix}0\\0\\2\\1\end{bmatrix} = 2 \tag{1.18}$$

and $\mathcal{P}\mathrm{v}$ is indeed positive, indicating that the point is outside the plane in the direction of the outward pointing normal. A point $\mathrm{v} = [0, 0, 0, 1]$ lies below the plane

$$\begin{bmatrix}0 & 0 & 1 & -1\end{bmatrix}\begin{bmatrix}0\\0\\0\\1\end{bmatrix} = -1 \tag{1.19}$$

The plane $[0, 0, 0, 0]$ is undefined.

1.5 Transformations

A transformation of the space H is a 4x4 matrix and can represent translation, rotation, stretching, and perspective transformations. Given a point u, its transformation v is represented by the matrix product

$$\mathrm{v} = \mathrm{Hu} \tag{1.20}$$

The corresponding plane transformation $\mathcal{P}$ to $\mathcal{Q}$ is

$$\mathcal{Q} = \mathcal{P}\mathrm{H}^{-1} \tag{1.21}$$

as we require that the condition

$$\mathcal{Q}\mathrm{v} = \mathcal{P}\mathrm{u} \tag{1.22}$$

is invariant under all transformations. To verify this we substitute from Equations 1.20 and 1.21 into the left hand side of 1.22 and we obtain on the right hand side $\mathrm{H}^{-1}\mathrm{H}$ which is the identity matrix I

$$\mathcal{P}\mathrm{H}^{-1}\mathrm{Hu} = \mathcal{P}\mathrm{u} \tag{1.23}$$

1.6 Translation Transformation

The transformation H corresponding to a translation by a vector $a\mathbf{i} + b\mathbf{j} + c\mathbf{k}$ is

$$\mathrm{H} = \mathrm{Trans}(a, b, c) = \begin{bmatrix}1 & 0 & 0 & a\\0 & 1 & 0 & b\\0 & 0 & 1 & c\\0 & 0 & 0 & 1\end{bmatrix} \tag{1.24}$$

Given a vector $\mathbf{u} = [x, y, z, w]^T$ the transformed vector v is given by

$$\mathbf{v} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \tag{1.25}$$

$$\mathbf{v} = \begin{bmatrix} x + aw \\ y + bw \\ z + cw \\ w \end{bmatrix} = \begin{bmatrix} x/w + a \\ y/w + b \\ z/w + c \\ 1 \end{bmatrix} \tag{1.26}$$

The translation may also be interpreted as the addition of the two vectors $(x/w)\mathbf{i} + (y/w)\mathbf{j} + (z/w)\mathbf{k}$ and $a\mathbf{i} + b\mathbf{j} + c\mathbf{k}$.

Every element of a transformation matrix may be multiplied by a non-zero constant without changing the transformation, in the same manner as points and planes. Consider the vector $2\mathbf{i} + 3\mathbf{j} + 2\mathbf{k}$ translated by, or added to $4\mathbf{i} - 3\mathbf{j} + 7\mathbf{k}$

$$\begin{bmatrix} 6 \\ 0 \\ 9 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & -3 \\ 0 & 0 & 1 & 7 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \\ 2 \\ 1 \end{bmatrix} \tag{1.27}$$

If we multiply the transformation matrix elements by, say, -5 and the vector elements by 2, we obtain

$$\begin{bmatrix} -60 \\ 0 \\ -90 \\ -10 \end{bmatrix} = \begin{bmatrix} -5 & 0 & 0 & -20 \\ 0 & -5 & 0 & 15 \\ 0 & 0 & -5 & -35 \\ 0 & 0 & 0 & -5 \end{bmatrix} \begin{bmatrix} 4 \\ 6 \\ 4 \\ 2 \end{bmatrix} \tag{1.28}$$

which corresponds to the vector $[6, 0, 9, 1]^T$ as before. The point $[2, 3, 2, 1]^T$ lies in the plane $[1, 0, 0, -2]$

$$[1, 0, 0, -2] \begin{bmatrix} 2 \\ 3 \\ 2 \\ 1 \end{bmatrix} = 0 \tag{1.29}$$

The transformed point is, as we have already found, $[6, 0, 9, 1]^T$. We will now compute the transformed plane. The inverse of the transform is

$$\begin{bmatrix} 1 & 0 & 0 & -4 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & -7 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and the transformed plane

$$[1 \quad 0 \quad 0 \quad -6] = [1 \quad 0 \quad 0 \quad -2] \begin{bmatrix} 1 & 0 & 0 & -4 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & -7 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.30}$$

Once again the transformed point lies in the transformed plane

$$[1 \quad 0 \quad 0 \quad -6] \begin{bmatrix} 6 \\ 0 \\ 9 \\ 1 \end{bmatrix} = 0 \tag{1.31}$$

1.7 Rotation Transformations

The transformation corresponding to rotations about the x, y, or z axes by an angle θ are

$$\mathrm{Rot}(x, \theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.32}$$

$$\mathrm{Rot}(y, \theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.33}$$

$$\mathrm{Rot}(z, \theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.34}$$

Let us interpret these rotations by means of an example. Given a point $\mathbf{u} = 7\mathbf{i} + 3\mathbf{j} + 2\mathbf{k}$ what is the effect of rotating it 90° about the z axis to v? The transform is obtained from Equation 1.34 with $\sin\theta = 1$ and $\cos\theta = 0$.

$$\begin{bmatrix} -3 \\ 7 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 7 \\ 3 \\ 2 \\ 1 \end{bmatrix} \tag{1.35}$$

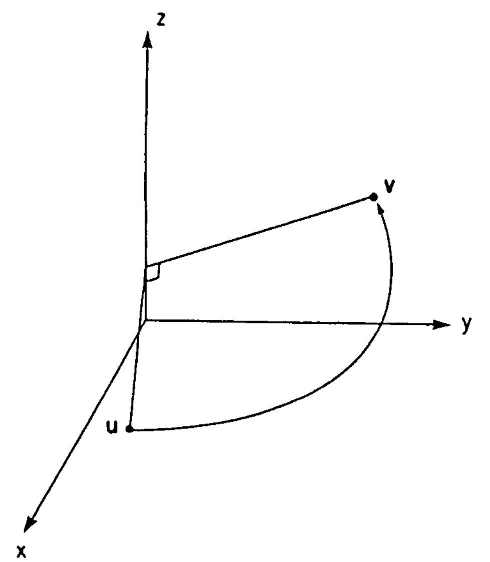


Figure 1.1. Rot(z, 90)

The initial and final points are shown in Figure 1.1. and it can be seen that the point has indeed been rotated 90° about the z axis. Let us now rotate $\mathbf{v}$ 90° about the y axis to $\mathbf{w}$. The transform is obtained from Equation 1.33 and we have

$$\begin{bmatrix} 2 \\ 7 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -3 \\ 7 \\ 2 \\ 1 \end{bmatrix} \tag{1.36}$$

This result is shown in Figure 1.2. If we combine these two rotations we have

$$\mathbf{v} = \mathrm{Rot}(z, 90)\mathbf{u} \tag{1.37}$$

and

$$\mathbf{w} = \mathrm{Rot}(y, 90)\mathbf{v} \tag{1.38}$$

Substituting for $\mathbf{v}$ from Equation 1.37 into Equation 1.38 we obtain

$$\mathbf{w} = \mathrm{Rot}(y, 90)\,\mathrm{Rot}(z, 90)\,\mathbf{u} \tag{1.39}$$

$$\mathrm{Rot}(y, 90)\,\mathrm{Rot}(z, 90) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.40}$$

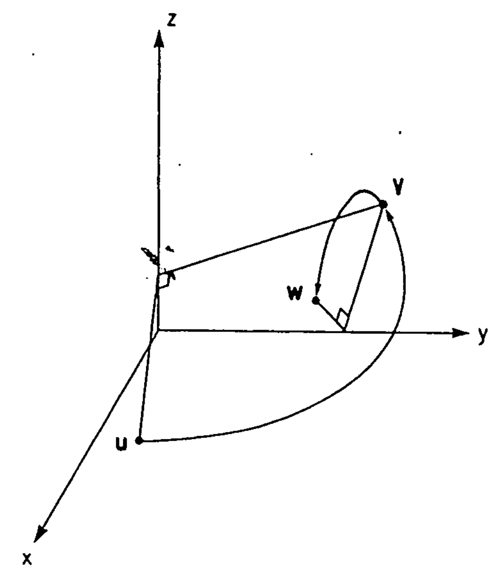


Figure 1.2. Rot(y, 90)

$$\mathrm{Rot}(y, 90)\,\mathrm{Rot}(z, 90) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.41}$$

thus

$$\mathbf{w} = \begin{bmatrix} 2 \\ 7 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 7 \\ 3 \\ 2 \\ 1 \end{bmatrix} \tag{1.42}$$

as we obtained before.

If we reverse the order of rotations and first rotate 90° about the y axis and then 90° about the z axis, we obtain a different position

$$\mathrm{Rot}(z, 90)\,\mathrm{Rot}(y, 90) = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.43}$$

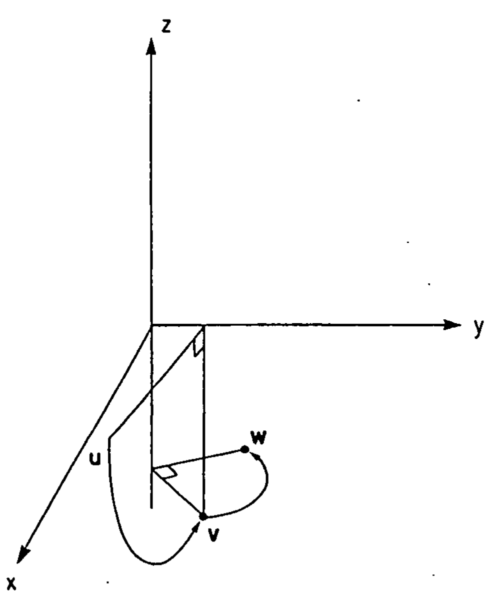


Figure 1.3. Rot(z, 90)Rot(y, 90)

and the point u transforms into w as

$$\begin{bmatrix} -3 \\ 2 \\ -7 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 7 \\ 3 \\ 2 \\ 1 \end{bmatrix} \tag{1.44}$$

We should expect this, as matrix multiplication is noncommutative.

$$AB \neq BA \tag{1.45}$$

The results of this transformation are shown in Figure 1.3.

We will now combine the original rotation with a translation $4i - 3j + 7k$. We obtain the translation from Equation 1.27 and the rotation from Equation 1.41. The matrix expression is

$$\text{Trans}(4, -3, 7)\ \text{Rot}(y, 90)\ \text{Rot}(z, 90) = \begin{bmatrix} 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & -3 \\ 0 & 0 & 1 & 7 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
$$= \begin{bmatrix} 0 & 0 & 1 & 4 \\ 1 & 0 & 0 & -3 \\ 0 & 1 & 0 & 7 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.46}$$

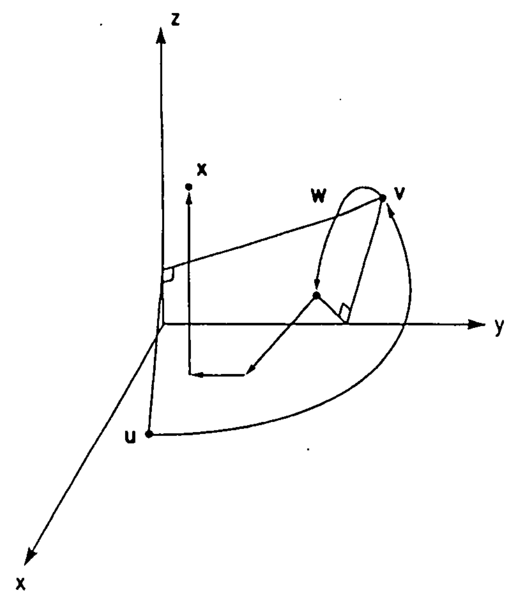


Figure 1.4. Trans(4, −3, 7)Rot(y, 90)Rot(z, 90)

and our point $w = 7i + 3j + 2k$ transforms into x as

$$\begin{bmatrix} 6 \\ 4 \\ 10 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 4 \\ 1 & 0 & 0 & -3 \\ 0 & 1 & 0 & 7 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 7 \\ 3 \\ 2 \\ 1 \end{bmatrix} \tag{1.47}$$

The result is shown in Figure 1.4.

1.8 Coordinate Frames

We can interpret the elements of the homogeneous transformation as four vectors describing a second coordinate frame. The vector $[0, 0, 0, 1]^T$ lies at the origin of the second coordinate frame. Its transformation corresponds to the right hand column of the transformation matrix. Consider the transform in Equation 1.47

$$\begin{bmatrix} 4 \\ -3 \\ 7 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 4 \\ 1 & 0 & 0 & -3 \\ 0 & 1 & 0 & 7 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{1.48}$$

The transform of the null vector is $[4, -3, 7, 1]^T$, the right hand column. If we transform vectors corresponding to unit vectors along the x, y, and z axes, we

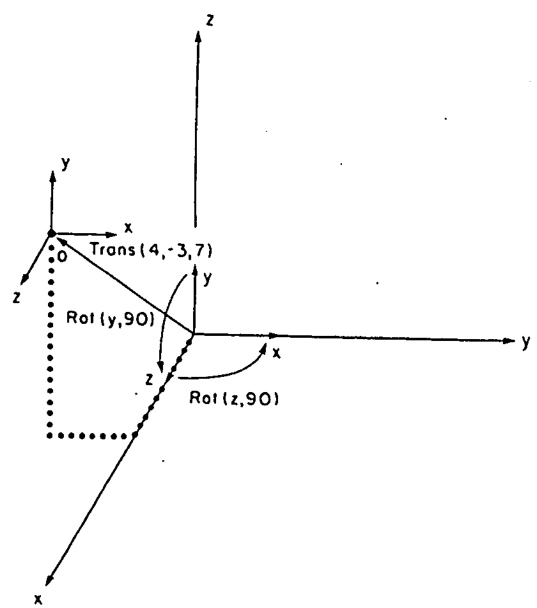


Figure 1.5. The Interpretation of a Transform as a Frame

obtain $[4, -2, 7, 1]^T$, $[4, -3, 8, 1]^T$, and $[5, -3, 7, 1]^T$, respectively. These four vectors are plotted in Figure 1.5 and form a coordinate frame.

The direction of these unit vectors is formed by subtracting the vector representing the origin of this coordinate frame and extending the vectors to infinity by reducing their scale factors to zero. The direction of the x, y, and z axes of this frame are $[0, 1, 0, 0]^T$, $[0, 0, 1, 0]^T$, and $[1, 0, 0, 0]^T$, respectively. These direction vectors correspond to the first three columns of the transformation matrix. The transformation matrix thus describes the three axis directions and the position of the origin of a coordinate frame rotated and translated away from the reference coordinate frame (see Figure 1.4). When a vector is transformed, as in Equation 1.47, the original vector can be considered as a vector described in the coordinate frame. The transformed vector is the same vector described with respect to the reference coordinate frame (see Figure 1.6).

1.9 Relative Transformations

The rotations and translations we have been describing have all been made with respect to the fixed reference coordinate frame. Thus, in the example given,

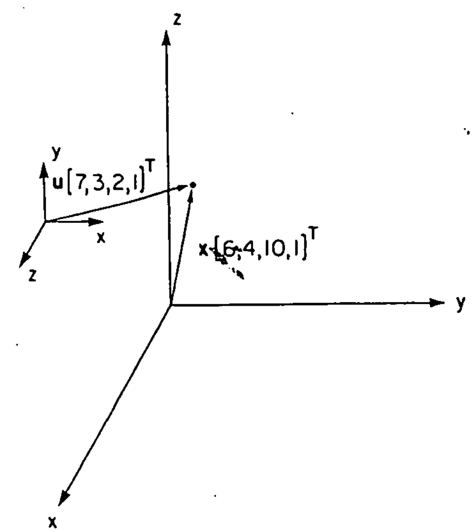


Figure 1.6. The Transformation of a Vector

$$\mathrm{Trans}(4, -3, 7)\ \mathrm{Rot}(y, 90)\ \mathrm{Rot}(z, 90) = \begin{bmatrix} 0 & 0 & 1 & 4 \\ 1 & 0 & 0 & -3 \\ 0 & 1 & 0 & 7 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.49}$$

the frame is first rotated around the reference z axis by 90°, then rotated 90° around the reference y axis, and finally translated by $4\mathbf{i} - 3\mathbf{j} + 7\mathbf{k}$, as shown in Figure 1.5. We may also interpret the operation in the reverse order, from left to right, as follows: the object is first translated by $4\mathbf{i} - 3\mathbf{j} + 7\mathbf{k}$; it is then rotated 90° around the current frame axes, which in this case are the same as the reference axes; it is then rotated 90° about the newly rotated (current) frame axes (see Figure 1.7).

In general, if we postmultiply a transform representing a frame by a second transformation describing a rotation and/or translation, we make that translation and/or rotation with respect to the frame axes described by the first transformation. If we premultiply the frame transformation by a transformation representing a translation and/or rotation, then that translation and/or rotation is made with respect to the base reference coordinate frame. Thus, given a frame C and a transformation T, corresponding to a rotation of 90° about the z axis, and a translation of 10 units in the x direction, we obtain a new position X when the change is made in base coordinates $\mathbf{X} = \mathbf{T}\,\mathbf{C}$

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 20 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 & 10 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 20 \\ 0 & 0 & -1 & 10 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.50}$$

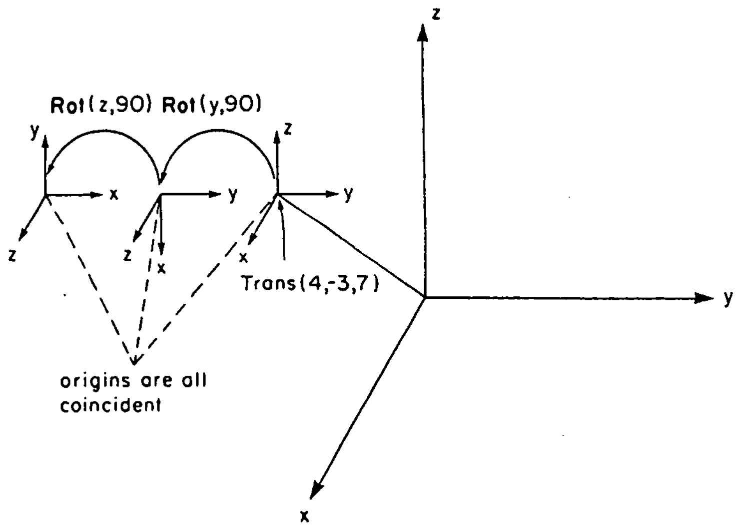


Figure 1.7. Relative Transformations

and a new position Y when the change is made relative to the frame axes as Y = C T

$$\begin{bmatrix} 0 & -1 & 0 & 30 \\ 0 & 0 & -1 & 10 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 20 \\ 0 & 0 & -1 & 10 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 & 10 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.51)$$

The results are shown in Figure 1.8.

1.10 Objects

Transformations are used to describe the position and orientation of objects. An object shown in Figure 1.9 is described by six points with respect to a coordinate frame fixed in the object.

If we rotate the object 90° about the z axis and then 90° about the y axis, followed by a translation of four units in the x direction, we can describe the transformation as

$$\mathrm{Trans}(4,0,0)\,\mathrm{Rot}(y,90)\,\mathrm{Rot}(z,90) = \begin{bmatrix} 0 & 0 & 1 & 4 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.52)$$

The transformation matrix represents the operation of rotation and translation on a coordinate frame originally aligned with the reference coordinate frame. We

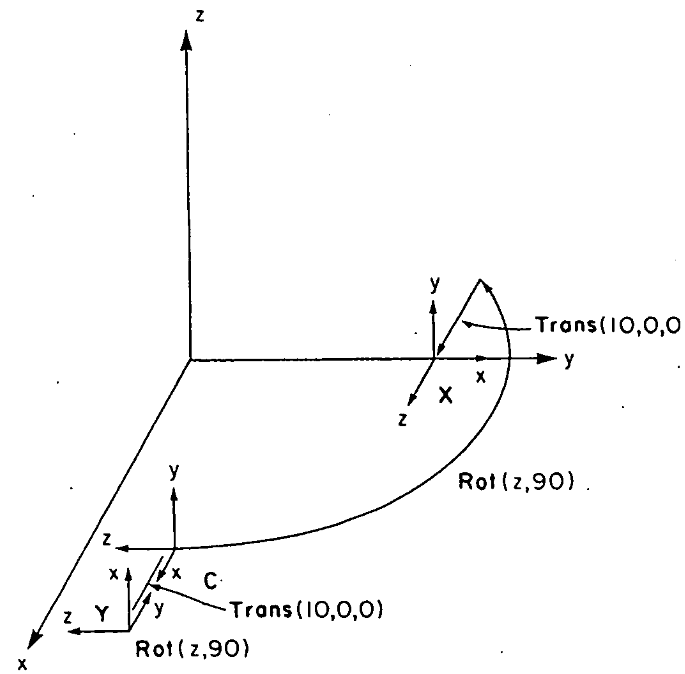


Figure 1.8. Transformations with Respect to Base and Frame Coordinates

may transform the six points of the object as

$$\begin{bmatrix} 4 & 4 & 6 & 6 & 4 & 4 \\ 1 & -1 & -1 & 1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 4 & 4 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 4 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -1 & -1 & 1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 4 & 4 \\ 0 & 0 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (1.53)$$

The results are plotted in Figure 1.10.

It can be seen that the object described in the figure bears the same fixed relationship to its coordinate frame, whose position and orientation are described by the transformation. Given an object described by a reference coordinate frame as in Figure 1.9, and a transformation representing the position and orientation of the object's axes, the object can be simply reconstructed, without the necessity of transforming all the points, by noting the direction and orientation of key features with respect to the describing frame's coordinate axes. By drawing the transformed coordinate frame, the object can be related to the new axis directions. In the example given, the long axis of the wedge lies along the y axis of the describing frame and, as the transformed y axis is in the z direction, the long axis will be upright in the transformed state, etc.

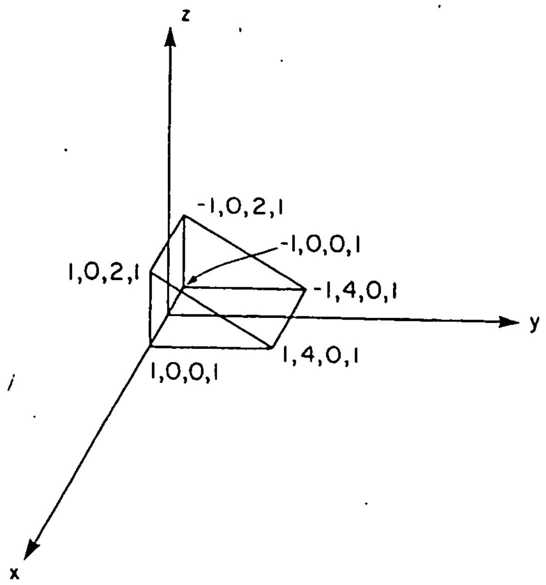


Figure 1.9. An Object

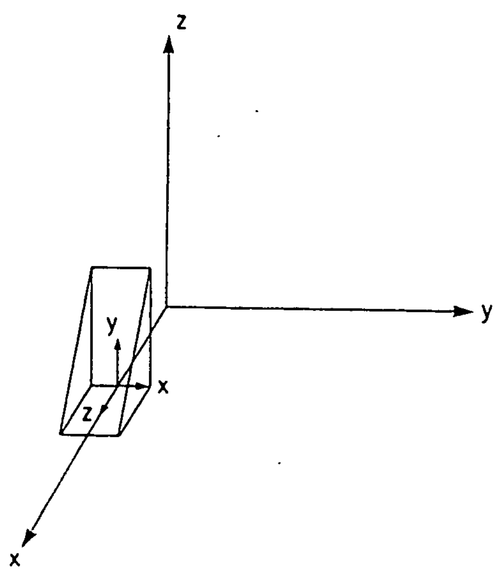


Figure 1.10. Transformed Wedge

1.11 Inverse Transformations

We are now in a position to develop the inverse transformation as the transform which carries the transformed coordinate frame back to the original frame. This is simply the description of the reference coordinate frame with respect to the transformed frame. Consider the example given in Figure 1.10. The direction of the reference frame x axis is $[0, 0, 1, 0]^T$ with respect to the transformed frame. The y and z axes are $[1, 0, 0, 0]^T$ and $[0, 1, 0, 0]^T$, respectively. The location of the origin is $[0, 0, -4, 1]^T$ with respect to the transformed frame and thus the inverse transform is

$$T^{-1} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & -4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.54}$$

That this is indeed the transform inverse is easily verified by multiplying it by the transform T to obtain an identity transform

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & -4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 4 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.55}$$

In general, given a transform with elements

$$T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.56}$$

then the inverse is

$$T^{-1} = \begin{bmatrix} n_x & n_y & n_z & -\mathbf{p}\cdot\mathbf{n} \\ o_x & o_y & o_z & -\mathbf{p}\cdot\mathbf{o} \\ a_x & a_y & a_z & -\mathbf{p}\cdot\mathbf{a} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.57}$$

where p, n, o, and a are the four column vectors and "·" represents the vector dot product. This result is easily verified by postmultiplying Equation 1.56 by Equation 1.57.

1.12 General Rotation Transformation

We stated the rotation transformations for rotations about the x, y, and z axes (Equations 1.32, 1.33, and 1.34). These transformations have a simple geometric

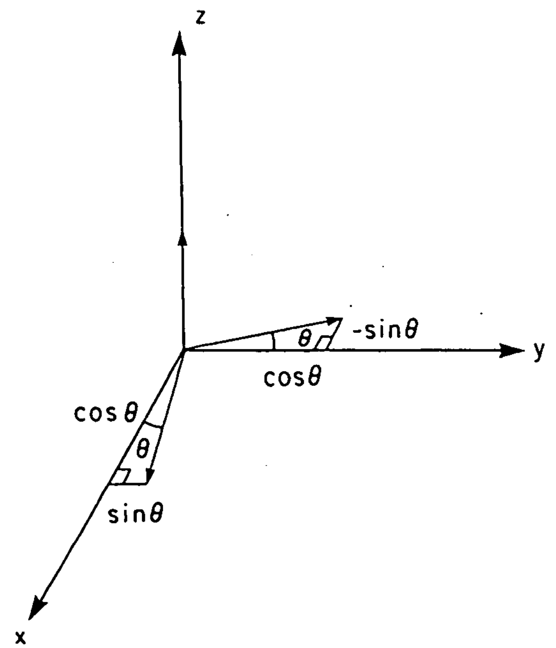


Figure 1.11. A Rotation about the z axis

interpretation. For example, in the case of a rotation about the z axis, the column representing the z axis will remain constant, while the column elements representing the x and y axes will vary as shown in Figure 1.11.

We will now develop the transformation matrix representing a rotation around an arbitrary vector k located at the origin. (See [Hamilton] for a full discussion of this subject.) In order to do this we will imagine that k is the z axis unit vector of a coordinate frame C

$$C = \begin{bmatrix} n_x & o_x & a_x & 0 \\ n_y & o_y & a_y & 0 \\ n_z & o_z & a_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.58}$$

$$k = a_x i + a_y j + a_z k \tag{1.59}$$

Rotating around the vector k is then equivalent to rotating around the z axis of the frame C.

$$\mathrm{Rot}(k, \theta) = \mathrm{Rot}({}^{C}z, \theta) \tag{1.60}$$

If we are given a frame T described with respect to the reference coordinate frame, we can find a frame X which describes the same frame with respect to

frame C as

$$T = C\,X \tag{1.61}$$

where X describes the position of T with respect to frame C. Solving for X we obtain

$$X = C^{-1}\,T \tag{1.62}$$

Rotating T around k is equivalent to rotating X around the z axis of frame C

$$\mathrm{Rot}(k, \theta)\,T = C\,\mathrm{Rot}(z, \theta)\,X \tag{1.63}$$

$$\mathrm{Rot}(k, \theta)\,T = C\,\mathrm{Rot}(z, \theta)\,C^{-1}\,T. \tag{1.64}$$

Thus

$$\mathrm{Rot}(k, \theta) = C\,\mathrm{Rot}(z, \theta)\,C^{-1} \tag{1.65}$$

However, we have only k, the z axis of the frame C. By expanding Equation 1.65 we will discover that $C\,\mathrm{Rot}(z, \theta)\,C^{-1}$ is a function of k only.

Multiplying $\mathrm{Rot}(z, \theta)$ on the right by C^{-1} we obtain

$$\mathrm{Rot}(z, \theta)\,C^{-1} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n_x & n_y & n_z & 0 \\ o_x & o_y & o_z & 0 \\ a_x & a_y & a_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} n_x\cos\theta - o_x\sin\theta & n_y\cos\theta - o_y\sin\theta & n_z\cos\theta - o_z\sin\theta & 0 \\ n_x\sin\theta + o_x\cos\theta & n_y\sin\theta + o_y\cos\theta & n_z\sin\theta + o_z\cos\theta & 0 \\ a_x & a_y & a_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.66}$$

premultiplying by

$$C = \begin{bmatrix} n_x & o_x & a_x & 0 \\ n_y & o_y & a_y & 0 \\ n_z & o_z & a_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.67}$$

we obtain C $\mathrm{Rot}(z,\theta)\,\mathrm{C}^{-1} =$

$$\begin{bmatrix} n_x n_x \cos\theta - n_x o_x \sin\theta + n_x o_x \sin\theta + o_x o_x \cos\theta + a_x a_x & n_x n_y \cos\theta - n_x o_y \sin\theta + n_y o_x \sin\theta + o_y o_x \cos\theta + a_x a_y & n_x n_z \cos\theta - n_x o_z \sin\theta + n_z o_x \sin\theta + o_z o_x \cos\theta + a_x a_z & 0 \\ n_y n_x \cos\theta - n_y o_x \sin\theta + n_x o_y \sin\theta + o_y o_x \cos\theta + a_y a_x & n_y n_y \cos\theta - n_y o_y \sin\theta + n_y o_y \sin\theta + o_y o_y \cos\theta + a_y a_y & n_y n_z \cos\theta - n_y o_z \sin\theta + n_z o_y \sin\theta + o_z o_y \cos\theta + a_y a_z & 0 \\ n_z n_x \cos\theta - n_z o_x \sin\theta + n_x o_z \sin\theta + o_z o_x \cos\theta + a_z a_x & n_z n_y \cos\theta - n_z o_y \sin\theta + n_y o_z \sin\theta + o_y o_z \cos\theta + a_z a_y & n_z n_z \cos\theta - n_z o_z \sin\theta + n_z o_z \sin\theta + o_z o_z \cos\theta + a_z a_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.68}$$

Simplifying, using the following relationships:
the dot product of any row or column of C with any other row or column is zero, as the vectors are orthogonal;
the dot product of any row or column of C with itself is 1 as the vectors are of unit magnitude;
the z unit vector is the vector cross product of the x and y vectors or

$$\mathbf{a} = \mathbf{n} \times \mathbf{o} \tag{1.69}$$

which has components

$$a_x = n_y o_z - n_z o_y$$
$$a_y = n_z o_x - n_x o_z.$$
$$a_z = n_x o_y - n_y o_x$$

the versine, abbreviated as $\mathrm{vers}\,\theta$, is defined as $\mathrm{vers}\,\theta = (1 - \cos\theta)$,
$k_x = a_x$, $k_y = a_y$, and $k_z = a_z$.
We obtain $Rot(k,\theta) =$

$$\begin{bmatrix} k_x k_x \,\mathrm{vers}\,\theta + \cos\theta & k_y k_x \,\mathrm{vers}\,\theta - k_z \sin\theta & k_z k_x \,\mathrm{vers}\,\theta + k_y \sin\theta & 0 \\ k_x k_y \,\mathrm{vers}\,\theta + k_z \sin\theta & k_y k_y \,\mathrm{vers}\,\theta + \cos\theta & k_z k_y \,\mathrm{vers}\,\theta - k_x \sin\theta & 0 \\ k_x k_z \,\mathrm{vers}\,\theta - k_y \sin\theta & k_y k_z \,\mathrm{vers}\,\theta + k_x \sin\theta & k_z k_z \,\mathrm{vers}\,\theta + \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.70}$$

This is an important result and should be thoroughly understood before proceeding further.

From this general rotation transformation we can obtain each of the elementary rotation transforms. For example $\mathrm{Rot}(x,\theta)$ is $\mathrm{Rot}(\mathbf{k},\theta)$ where $k_x = 1$, $k_y = 0$,

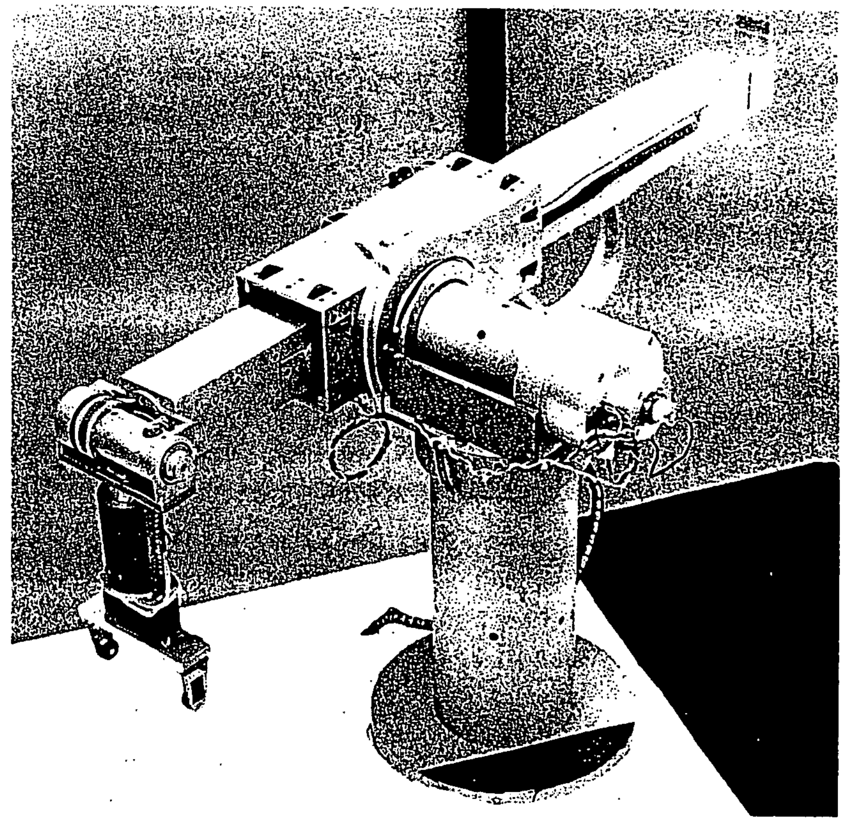


Figure 1.12. The Stanford Manipulator

and $k_z = 0$. Substituting these values of k into Equation 1.70 we obtain

$$\mathrm{Rot}(x,\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.71}$$

as before.

1.13 Equivalent Angle and Axis of Rotation

Given any arbitrary rotational transformation, we can use Equation 1.70 to obtain an axis about which an equivalent rotation θ is made as follows. Given a rota-

tional transformation R

$$R = \begin{bmatrix} n_x & o_x & a_x & 0 \\ n_y & o_y & a_y & 0 \\ n_z & o_z & a_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.72)$$

we may equate R to Rot(k, θ)

$$\begin{bmatrix} n_x & o_x & a_x & 0 \\ n_y & o_y & a_y & 0 \\ n_z & o_z & a_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$
$$\begin{bmatrix} k_x k_x \operatorname{vers}\theta + \cos\theta & k_y k_x \operatorname{vers}\theta - k_z \sin\theta & k_z k_x \operatorname{vers}\theta + k_y \sin\theta & 0 \\ k_x k_y \operatorname{vers}\theta + k_z \sin\theta & k_y k_y \operatorname{vers}\theta + \cos\theta & k_z k_y \operatorname{vers}\theta - k_x \sin\theta & 0 \\ k_x k_z \operatorname{vers}\theta - k_y \sin\theta & k_y k_z \operatorname{vers}\theta + k_x \sin\theta & k_z k_z \operatorname{vers}\theta + \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.73)$$

Summing the diagonal terms of Equation 1.73 we obtain

$$n_x + o_y + a_z + 1 = k_x^2 \operatorname{vers}\theta + \cos\theta + k_y^2 \operatorname{vers}\theta + \cos\theta + k_z^2 \operatorname{vers}\theta + \cos\theta + 1 \quad (1.74)$$

$$\begin{aligned} n_x + o_y + a_z &= (k_x^2 + k_y^2 + k_z^2) \operatorname{vers}\theta + 3\cos\theta \\ &= 1 + 2\cos\theta \end{aligned} \quad (1.75)$$

and the cosine of the angle of rotation is

$$\cos\theta = \frac{1}{2}(n_x + o_y + a_z - 1) \quad (1.76)$$

Differencing pairs of off-diagonal terms in Equation 1.73 we obtain

$$o_z - a_y = 2k_x \sin\theta \quad (1.77)$$
$$a_x - n_z = 2k_y \sin\theta \quad (1.78)$$
$$n_y - o_x = 2k_z \sin\theta \quad (1.79)$$

Squaring and adding Equations 1.77 – 1.79 we obtain an expression for $\sin\theta$

$$(o_z - a_y)^2 + (a_x - n_z)^2 + (n_y - o_x)^2 = 4\sin^2\theta \quad (1.80)$$

and the sine of the angle of rotation is

$$\sin\theta = \pm\frac{1}{2}\sqrt{(o_z - a_y)^2 + (a_x - n_z)^2 + (n_y - o_x)^2} \quad (1.81)$$

We may define the rotation to be positive about the vector k such that $0 \leq \theta \leq 180°$. In this case the + sign is appropriate in Equation 1.81 and thus the angle of rotation θ is uniquely defined as

$$\tan\theta = \frac{\sqrt{(o_z - a_y)^2 + (a_x - n_z)^2 + (n_y - o_x)^2}}{(n_x + o_y + a_z - 1)} \quad (1.82)$$

The components of k may be obtained from Equations 1.77 – 1.79 as

$$k_x = \frac{o_z - a_y}{2\sin\theta} \quad (1.83)$$

$$k_y = \frac{a_x - n_z}{2\sin\theta} \quad (1.84)$$

$$k_z = \frac{n_y - o_x}{2\sin\theta} \quad (1.85)$$

When the angle of rotation is very small, the axis of rotation is physically not well defined due to the small magnitude of both numerator and denominator in Equations 1.83–1.85. If the resulting angle is small, the vector k should be renormalized to ensure that $|k| = 1$. When the angle of rotation approaches 180° the vector k is once again poorly defined by Equation 1.83–1.85 as the magnitude of the sine is again decreasing [Klump]. The axis of rotation is, however, physically well defined in this case. When $\theta > 150°$, the denominator of Equations 1.83-1.85 is less than 1. As the angle increases to 180° the rapidly decreasing magnitude of both numerator and denominator leads to considerable inaccuracies in the determination of k. At $\theta = 180°$, Equations 1.83–1.85 are of the form 0/0, yielding no information at all about a physically well defined vector k. If the angle of rotation is greater than 90°, then we must follow a different approach in determining k. Equating the diagonal elements of Equation 1.73 we obtain

$$k_x^2 \operatorname{vers}\theta + \cos\theta = n_x \quad (1.86)$$
$$k_y^2 \operatorname{vers}\theta + \cos\theta = o_y \quad (1.87)$$
$$k_z^2 \operatorname{vers}\theta + \cos\theta = a_z \quad (1.88)$$

Substituting for $\cos\theta$ and $\operatorname{vers}\theta$ from Equation 1.76 and solving for the elements of k we obtain further

$$k_x = \pm\sqrt{\frac{n_x - \cos\theta}{1 - \cos\theta}} \quad (1.89)$$

$$k_y = \pm\sqrt{\frac{o_y - \cos\theta}{1 - \cos\theta}} \quad (1.90)$$

$$k_z = \pm\sqrt{\frac{a_z - \cos\theta}{1 - \cos\theta}} \quad (1.91)$$

The largest component of k defined by Equations 1.89 – 1.91 corresponds to the most positive component of n_x, o_y, and a_z. For this largest element, the sign of the radical can be obtained from Equations 1.77 – 1.79. As the sine of the angle of rotation θ must be positive, then the sign of the component of k defined by Equations 1.77 – 1.79 must be the same as the sign of the left hand side of these equations. Thus we may combine Equations 1.89–1.91 with the information contained in Equations 1.77–1.79 as follows

$$k_x = \operatorname{sgn}(o_z - a_y)\sqrt{\frac{n_x - \cos\theta}{1 - \cos\theta}} \tag{1.92}$$

$$k_y = \operatorname{sgn}(a_x - n_z)\sqrt{\frac{o_y - \cos\theta}{1 - \cos\theta}} \tag{1.93}$$

$$k_z = \operatorname{sgn}(n_y - o_x)\sqrt{\frac{a_z - \cos\theta}{1 - \cos\theta}} \tag{1.94}$$

where $\operatorname{sgn}(e) = +1$ if $e \geq 0$ and $\operatorname{sgn}(e) = -1$ if $e \leq 0$.

Only the largest element of k is determined from Equations 1.92–1.94, corresponding to the most positive element of n_x, o_y, and a_z. The remaining elements are more accurately determined by the following equations formed by summing pairs of off-diagonal elements of Equation 1.73

$$n_y + o_x = 2k_x k_y \operatorname{vers}\theta \tag{1.95}$$

$$o_z + a_y = 2k_y k_z \operatorname{vers}\theta \tag{1.96}$$

$$n_z + a_x = 2k_z k_x \operatorname{vers}\theta \tag{1.97}$$

If k_x is largest then

$$k_y = \frac{n_y + o_x}{2k_x \operatorname{vers}\theta} \qquad \text{from Equation 1.95} \tag{1.98}$$

$$k_z = \frac{a_x + n_z}{2k_x \operatorname{vers}\theta} \qquad \text{from Equation 1.97} \tag{1.99}$$

If k_y is largest then

$$k_x = \frac{n_y + o_x}{2k_y \operatorname{vers}\theta} \qquad \text{from Equation 1.95} \tag{1.100}$$

$$k_z = \frac{o_z + a_y}{2k_y \operatorname{vers}\theta} \qquad \text{from Equation 1.96} \tag{1.101}$$

If k_z is largest then

$$k_x = \frac{a_x + n_z}{2k_z \operatorname{vers}\theta} \qquad \text{from Equation 1.97} \tag{1.102}$$

$$k_y = \frac{o_z + a_y}{2k_z \operatorname{vers}\theta} \qquad \text{from Equation 1.96} \tag{1.103}$$

(See [Whitney] for an alternate approach to this problem.)

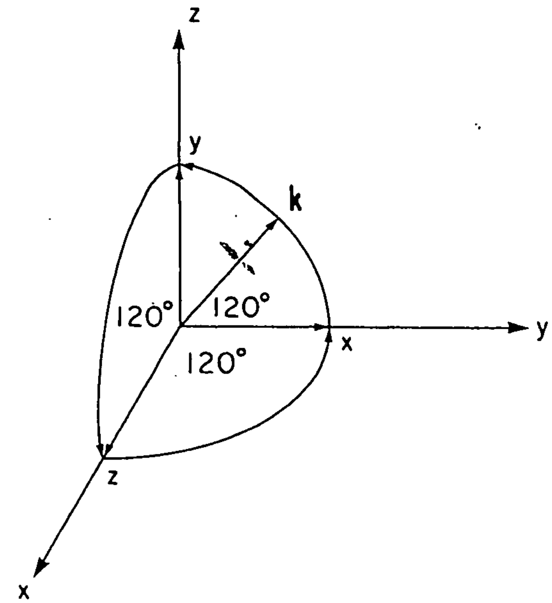


Figure 1.13. Rotation about k of 120°

Example 1.1

Determine the equivalent axis and angle of rotation for the matrix given in Equation 1.41

$$\operatorname{Rot}(y,90)\operatorname{Rot}(z,90) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.104}$$

We first determine $\cos\theta$ from Equation 1.76

$$\cos\theta = \frac{1}{2}(0 + 0 + 0 - 1) = -\frac{1}{2} \tag{1.105}$$

and $\sin\theta$ from Equation 1.81

$$\sin\theta = \frac{1}{2}\sqrt{(1-0)^2 + (1-0)^2 + (1-0)^2} = \frac{\sqrt{3}}{2} \tag{1.106}$$

Thus

$$\theta = \tan^{-1}\left(\frac{\sqrt{3}}{2} \Big/ \frac{-1}{2}\right) = 120^\circ \tag{1.107}$$

As $\theta > 90$, we determine the largest component of k corresponding to the largest element on the diagonal. As all diagonal elements are equal in this example we may pick any one. We will pick k_x given by Equation 1.92

$$k_x = +\sqrt{0 + \frac{1}{2} \Big/ 1 + \frac{1}{2}} = \frac{1}{\sqrt{3}} \tag{1.108}$$

As we have determined k_x we may now determine k_y and k_z from Equations 1.98 and 1.99, respectively

$$k_y = \frac{1+0}{\sqrt{3}} = \frac{1}{\sqrt{3}} \tag{1.109}$$

$$k_z = \frac{1+0}{\sqrt{3}} = \frac{1}{\sqrt{3}} \tag{1.110}$$

In summary, then

$$\mathrm{Rot}(y, 90)\mathrm{Rot}(z, 90) = \mathrm{Rot}(\mathrm{k}, 120) \tag{1.111}$$

where

$$\mathrm{k} = \frac{1}{\sqrt{3}}\mathrm{i} + \frac{1}{\sqrt{3}}\mathrm{j} + \frac{1}{\sqrt{3}}\mathrm{k} \quad \text{(see Figure 1.13)} \tag{1.112}$$

Any combination of rotations is always equivalent to a single rotation about some axis k by an angle θ, an important result that we will make use of later.

1.14 Stretching and Scaling

Although we will not use these deforming transformations in manipulation, we include them here to complete the subject of transformations.

A transform T

$$\mathrm{T} = \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.113}$$

will stretch objects uniformly along the x axis by a factor a, along the y axis by a factor b, and along the z axis by a factor c. Consider any point on an object $x\mathrm{i} + y\mathrm{j} + z\mathrm{k}$; its transform is

$$\begin{bmatrix} ax \\ by \\ cz \\ 1 \end{bmatrix} = \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{1.114}$$

indicating stretching as stated. Thus a cube could be transformed into a rectangular parallelepiped by such a transform.

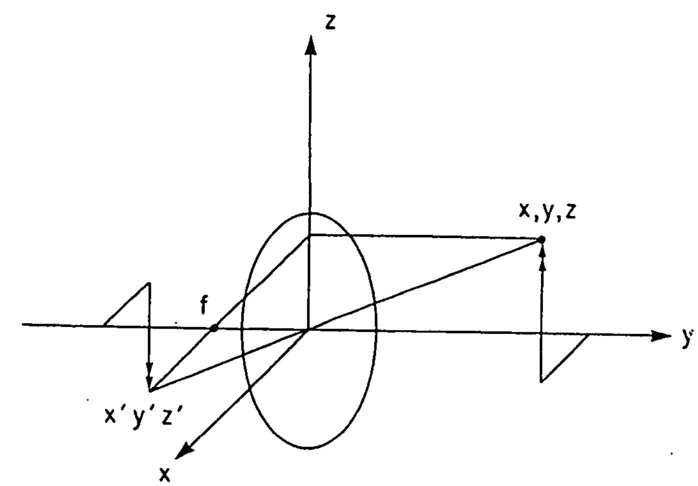


Figure 1.14. Perspective Transformation

The transform S where

$$\mathrm{S} = \begin{bmatrix} s & 0 & 0 & 0 \\ 0 & s & 0 & 0 \\ 0 & 0 & s & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.115}$$

will scale any object by the factor s.

1.15 Perspective Transformations

Consider the image formed of an object by a simple lens as shown in Figure 1.14.

The axis of the lens is shown along the y axis for convenience. An object point x, y, z is imaged at x' y' z' if the lens has a focal length f (f is considered positive). y' represents the image distance and varies with object distance y. If we plot points on a plane perpendicular to the y axis located at y' (the film plane in a camera), then a perspective image is formed.

We will first obtain values of x', y', and z', then introduce a perspective transformation and show that the same values are obtained.

Based on the fact that a ray passing through the center of the lens is undeviated, we may write

$$\frac{z}{y} = \frac{z'}{y'} \tag{1.116}$$

and

$$\frac{x}{y} = \frac{x'}{y'} \tag{1.117}$$

Based on the additional fact that a ray parallel to the lens axis passes through the focal point f, we may write

$$\frac{z}{f} = \frac{z'}{y'+f} \tag{1.118}$$

and

$$\frac{x}{f} = \frac{x'}{y'+f} \tag{1.119}$$

Notice that x', y', and z' are negative and that f is positive. Eliminating y' between Equations 1.116 and 1.118 we obtain

$$\frac{z}{f} = \frac{z'}{(\frac{z'y}{z}+f)} \tag{1.120}$$

and solving for z' we obtain the result

$$z' = \frac{z}{(1-\frac{y}{f})} \tag{1.121}$$

Working with Equations 1.117 and 1.119 we can similarly obtain

$$x' = \frac{x}{(1-\frac{y}{f})} \tag{1.122}$$

In order to obtain the image distance y' we rewrite Equations 1.116 and 1.118 as

$$\frac{z}{z'} = \frac{y}{y'} \tag{1.123}$$

and

$$\frac{z}{z'} = \frac{f}{y'+f} \tag{1.124}$$

thus

$$\frac{y}{y'} = \frac{f}{y'+f} \tag{1.125}$$

and solving for y' we obtain the result

$$y' = \frac{y}{(1-\frac{y}{f})} \tag{1.126}$$

The homogeneous transformation P which produces the same result is

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -\frac{1}{f} & 0 & 1 \end{bmatrix} \tag{1.127}$$

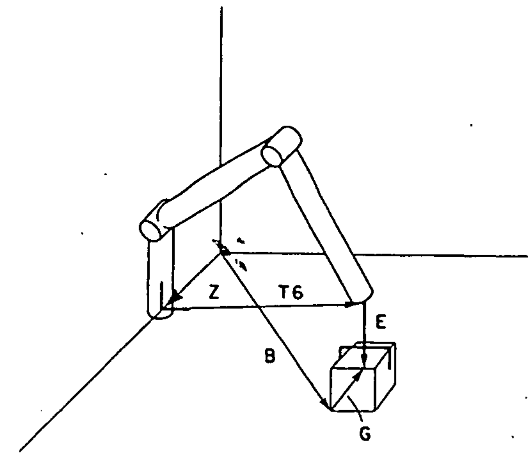


Figure 1.15. An Object and Manipulator

as any point $x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$ transforms as

$$\begin{bmatrix} x \\ y \\ z \\ 1-\frac{y}{f} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -\frac{1}{f} & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{1.128}$$

The image point x', y', z', obtained by dividing through by the weight factor $(1 - y/f)$, is

$$\frac{x}{(1-y/f)}\mathbf{i} + \frac{y}{(1-y/f)}\mathbf{j} + \frac{z}{(1-y/f)}\mathbf{k} \tag{1.129}$$

This is the same result that we obtained above.

A transform similar to P but with $-1/f$ at the bottom of the first column produces a perspective transformation along the x axis. If the $-1/f$ term is in the third column then the projection is along the z axis.

1.16 Transform Equations

We will frequently be required to deal with transform equations in which a coordinate frame is described in two or more ways. Consider the situation described in Figure 1.15. A manipulator is positioned with respect to base coordinates by a transform Z. The end of the manipulator is described by a transform ${}^{Z}\mathrm{T}_6$, and the end effector is described by ${}^{T_6}\mathrm{E}$. An object is positioned with respect to base coordinates by a transform B, and finally the manipulator end effector is positioned with respect to the object by ${}^{B}\mathrm{G}$. We have two descriptions of the position of the end effector, one with respect to the object and one with respect to the

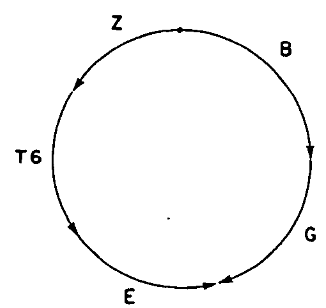


Figure 1.16. Directed Transform Graph

manipulator. As both positions are the same, we may equate the two descriptions

$$Z\,{}^{Z}T_6\,{}^{T_6}E = B\,{}^{B}G \tag{1.130}$$

This equation may be represented by the directed transform graph (see Figure 1.16). Each link of the graph represents a transform and is directed from its defining coordinate frame.

If we wish to solve Equation 1.130 for the manipulator transform T_6 we must premultiply Equation 1.130 by Z^{-1} and postmultiply by E^{-1} to obtain

$$T_6 = Z^{-1}\,B\,G\,E^{-1} \tag{1.131}$$

We may obtain this result from the transform graph by starting at the base of the T_6 link and listing the transforms as we traverse the graph in order until we reach the head of link T_6. If, as we list transforms, we move from a head to a tail (in the reverse direction of the directed link), we list the inverse of the transform. From the graph we thus obtain

$$T_6 = Z^{-1}\,B\,G\,E^{-1} \tag{1.132}$$

as before.

As a further example, consider that the position of the object B is unknown, but that the manipulator is moved such that the end effector is positioned over the object correctly. We may then solve for B from Equation 1.130 by postmultiplying by G^{-1} or obtain the same result directly from the graph by tracing the path from the tail of B back around the graph to the head of link B

$$B = Z\,T_6\,E\,G^{-1} \tag{1.133}$$

We may also use the graph to solve for connected groups of transforms, for example

$$Z\,T_6 = B\,G\,E^{-1} \tag{1.134}$$

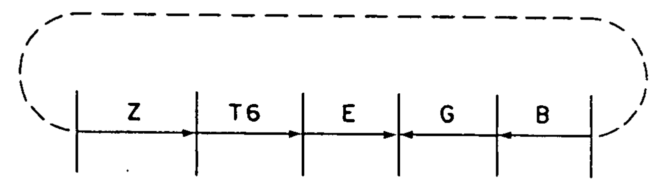


Figure 1.17. An Alternative Form of Figure 1.16

The use of a transform graph simplifies the solution of transform equations, allowing the results to be written directly. In order to avoid drawing circles, we will represent the transform graph as shown in Figure 1.17, where the dashed line indicates that two nodes are connected together.

Intermediate vertical lines represent individual coordinate frames.

1.17 Summary

Homogeneous transformations may be readily used to describe the positions and orientations of coordinate frames in space. If a coordinate frame is embedded in an object then the position and orientation of the object are also readily described.

The description of object A in terms of object B by means of a homogeneous transformation may be inverted to obtain the description of object B in terms of object A. This is not a property of a simple vector description of the relative displacement of one object with respect to another.

Transformations may be interpreted as a product of rotation and translation transformations. If they are interpreted from left to right, then the rotations and translations are in terms of the currently defined coordinate frame. If they are interpreted from right to left, then the rotations and translations are described with respect to the reference coordinate frame.

Homogeneous transformations describe coordinate frames in terms of rectangular components, which are the sines and cosines of angles. This description may be related to rotations in which case the description is in terms of a vector and angle of rotation.

1.18 References

Denavit, J. & Hartenberg, R. S. "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices," *ASME Journal of Applied Mechanics* (June 1955), 215–221.

Duda, R. O. & Hart, P. E. *Pattern Classification and Scene Analysis*, Wiley, New York, 1972.

Hamilton, W. R. *Elements of Quaternions*, Chelsea Publishing Co., New York, 1969.

Klumpp, A. R. "Singularity-Free Extraction of a Quaternion from a Direction Cosine Matrix," *Journal of Spacecraft* 13, 12 (Dec. 1976), 754–755.

Paul, R. P. Modeling, Trajectory Calculation and Servoing of a Computer Controlled Arm, Stanford Artificial Intelligence Laboratory, Stanford University, AIM 177, 1972.

Paul, R. P. "The Mathematics of Computer Controlled Manipulation," *The 1977 Joint Automatic Control Conference*, July 1977, 124–131.

Pieper, D. L. The Kinematics of Manipulators Under Computer Control, Stanford Artificial Intelligence Laboratory, Stanford University, AIM 72, 1968.

Roberts, L. G. Machine Perception of Three-Dimensional Solids, Lincoln Laboratory, Massachusetts Institute of Technology, Report No. 315, 1963.

Roberts, L. G. Homogeneous Matrix Representation and Manipulation of N-Dimensional Constructs, Lincoln Laboratory, Massachusetts Institute of Technology, Document No. MS1045, 1965.

Whitney, D. E. "The Mathematics of Coordinated Control of Prosthetic Arms and Manipulators," *Trans. ASME, Journal of Dynamic Systems, Measurement, and Control* (Dec. 1972), 303–309.

KINEMATIC EQUATIONS

2.1 Introduction

In this chapter we will develop homogeneous transformations to represent various coordinate frames and formulate methods of assigning coordinate frames to mechanical linkages representing manipulators. We will first define various methods of describing the position and orientation of a manipulator and then develop this description in terms of the joint coordinates.

Any manipulator can be considered to consist of a series of links connected together by joints. We will embed a coordinate frame in each link of the manipulator. Using homogeneous transformations, we can describe the relative position and orientation between these coordinate frames [Pieper]. Historically, the homogeneous transformation describing the relation between one link and the next has been called an A matrix [Denavit]. An A matrix is simply a homogeneous transformation describing the relative translation and rotation between link coordinate systems. A_1 describes the position and orientation of the first link. A_2 describes the position and orientation of the second link with respect to the first. Thus the position and orientation of the second link in base coordinates are given by the matrix product

$$T_2 = A_1 A_2 \tag{2.1}$$

Similarly, A_3 describes the third link in terms of the second and

$$T_3 = A_1 A_2 A_3 \tag{2.2}$$

These products of A matrices have historically been called T matrices, with the leading superscript omitted if it is 0. Given a six link manipulator we have

$$T_6 = A_1 A_2 A_3 A_4 A_5 A_6 \tag{2.3}$$

A six link manipulator can have six degrees of freedom, one for each link, and can be positioned and oriented arbitrarily within its range of motion. Three degrees of freedom are required to specify position and three more to specify orientation. T_6 represents the position and orientation of the manipulator. This can be thought of

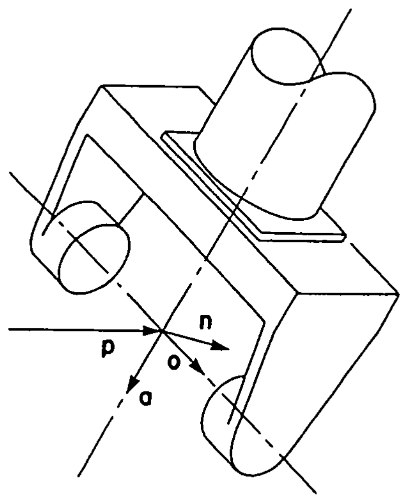


Figure 2.1. o, a, and p Vectors

in terms of a hand, as shown in Figure 2.1. We locate the origin of a describing coordinate frame centrally between the finger tips. This origin is described by a vector **p**.

The three unit vectors describing the hand orientation are directed as follows. The z vector lies in the direction from which the hand would approach an object and is known as the approach vector, **a**. The y vector, known as the orientation vector **o**, is in the direction specifying the orientation of the hand, from fingertip to fingertip. The final vector, known as the normal vector, **n**, forms a right-handed set of vectors and is thus specified by the vector cross-product

$$\mathbf{n} = \mathbf{o} \times \mathbf{a}$$

The transform T_6 thus has elements

$$T_6 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

as shown in Figure 2.1.

2.2 Specification of Orientation

T_6 is fully specified by assigning values to each of its 16 elements. Of these 16 elements only 12 have any real meaning. The bottom row consists of three zeros and a one. The left hand column vector is the vector cross product of the **o** and **a** column vectors. The remaining nine numbers represent three vectors **o**, **a**, and **p**. While there is no restriction on the value of **p**, provided the manipulator can reach the desired position, the vectors **o** and **a** must both be of unit magnitude and perpendicular, i.e.

$$\mathbf{o} \cdot \mathbf{o} = 1 \quad (2.5)$$

$$\mathbf{a} \cdot \mathbf{a} = 1 \quad (2.6)$$

$$\mathbf{o} \cdot \mathbf{a} = 0 \quad (2.7)$$

These restrictions on **o** and **a** make it difficult to assign components to the vectors except in simple cases when the end effector is aligned with the coordinate axes. If there is any doubt that **o** and **a** meet these conditions, the vectors may be modified to satisfy the conditions as follows. Make **a** of unit magnitude

$$\mathbf{a} \leftarrow \frac{\mathbf{a}}{|\mathbf{a}|} \quad (2.8)$$

construct **n** perpendicular to **o** and **a**

$$\mathbf{n} \leftarrow \mathbf{o} \times \mathbf{a} \quad (2.9)$$

rotate **o** in the plane formed by **o** and **a**, so that it is perpendicular to both **n** and **a**

$$\mathbf{o} \leftarrow \mathbf{a} \times \mathbf{n} \quad (2.10)$$

and make **o** of unit magnitude

$$\mathbf{o} \leftarrow \frac{\mathbf{o}}{|\mathbf{o}|} \quad (2.11)$$

We may also specify the orientation of the end of the manipulator as a rotation θ about an axis **k** using the generalized rotation matrix, $\mathrm{Rot}(\mathbf{k}, \theta)$, developed in Chapter 1. Unfortunately, the axis of rotation to achieve some desired orientations is not intuitively obvious.

2.3 Euler Angles

Orientation is more frequently specified by a sequence of rotations about the x, y, or z axes. Euler angles describe any possible orientation in terms of a rotation ϕ about the z axis, then a rotation θ about the new y axis, y', and finally, a rotation about the new z axis, z'', of ψ. (See Figure 2.2).

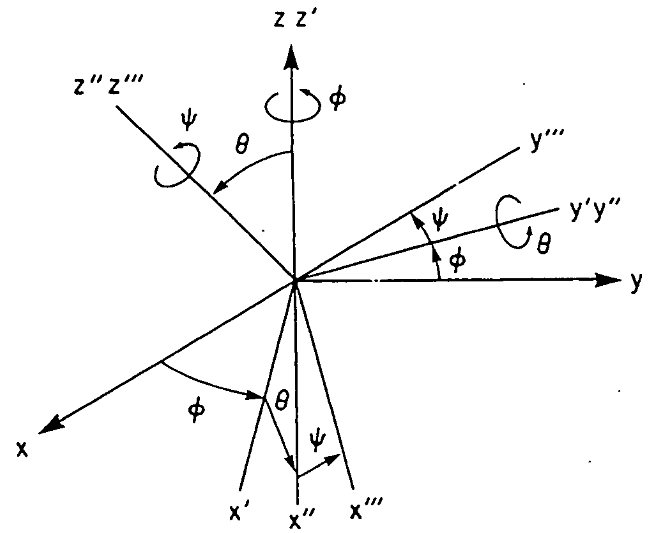


Figure 2.2. Euler Angles

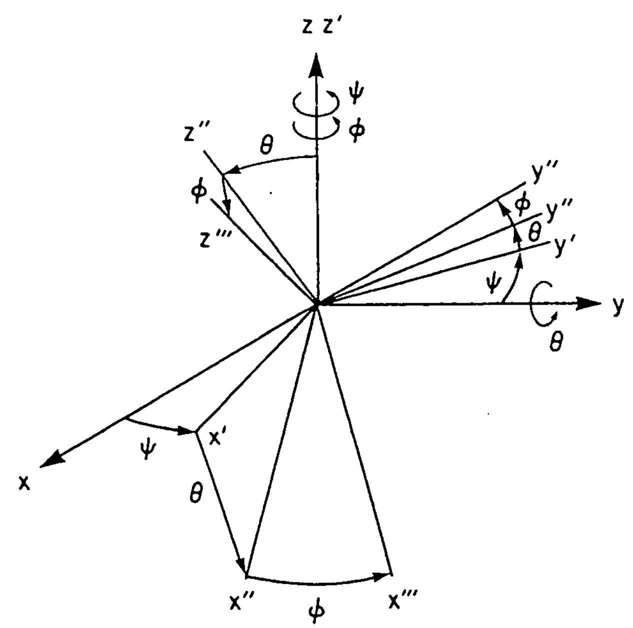


Figure 2.3. Euler Angles Interpreted in Base Coordinates

$$\mathrm{Euler}(\phi,\theta,\psi) = \mathrm{Rot}(z,\phi)\,\mathrm{Rot}(y,\theta)\,\mathrm{Rot}(z,\psi) \tag{2.12}$$

As in every case of a sequence of rotations, the order in which the rotations are made is important. Notice that this sequence of rotations can be interpreted in the reverse order as rotations in base coordinates: a rotation ψ about the z axis, followed by a rotation θ about the base y axis, and finally a rotation ϕ, once again about the base z axis (see Figure 2.3).

The Euler transformation, $\mathrm{Euler}(\phi,\theta,\psi)$, can be evaluated by multiplying the three rotation matrices together

$$\mathrm{Euler}(\phi,\theta,\psi) = \mathrm{Rot}(z,\phi)\,\mathrm{Rot}(y,\theta)\,\mathrm{Rot}(z,\psi) \tag{2.13}$$

$$\mathrm{Euler}(\phi,\theta,\psi) = \mathrm{Rot}(z,\phi)\begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} \cos\psi & -\sin\psi & 0 & 0 \\ \sin\psi & \cos\psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.14}$$

$$\mathrm{Euler}(\phi,\theta,\psi) = \begin{bmatrix} \cos\phi & -\sin\phi & 0 & 0 \\ \sin\phi & \cos\phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} \cos\theta\cos\psi & -\cos\theta\sin\psi & \sin\theta & 0 \\ \sin\psi & \cos\psi & 0 & 0 \\ -\sin\theta\cos\psi & \sin\theta\sin\psi & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.15}$$

$$\mathrm{Euler}(\phi,\theta,\psi) = \begin{bmatrix} \cos\phi\cos\theta\cos\psi - \sin\phi\sin\psi & -\cos\phi\cos\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\sin\theta & 0 \\ \sin\phi\cos\theta\cos\psi + \cos\phi\sin\psi & -\sin\phi\cos\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\sin\theta & 0 \\ -\sin\theta\cos\psi & \sin\theta\sin\psi & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.16}$$

2.4 Roll, Pitch, and Yaw

Another frequently used set of rotations is roll, pitch, and yaw.

If we imagine a ship steaming along the z axis, then roll corresponds to a rotation ϕ about the z axis, pitch corresponds to a rotation θ about the y axis, and yaw corresponds to a rotation ψ about the x axis (see Figure 2.4). The rotations applied to a manipulator end effector are shown in Figure 2.5.

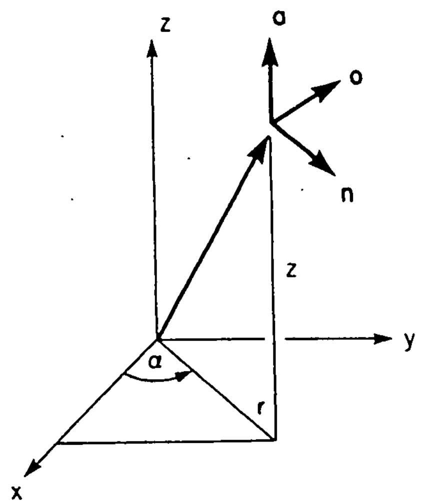


Figure 2.6. Cylindrical Polar Coordinates

$$\mathrm{Cyl}(z,\alpha,r)=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&z\\0&0&0&1\end{bmatrix}\begin{bmatrix}\cos\alpha&-\sin\alpha&0&r\cos\alpha\\\sin\alpha&\cos\alpha&0&r\sin\alpha\\0&0&1&0\\0&0&0&1\end{bmatrix}\tag{2.23}$$

$$\mathrm{Cyl}(z,\alpha,r)=\begin{bmatrix}\cos\alpha&-\sin\alpha&0&r\cos\alpha\\\sin\alpha&\cos\alpha&0&r\sin\alpha\\0&0&1&z\\0&0&0&1\end{bmatrix}\tag{2.24}$$

If we were to postmultiply this transformation by an orientation transform, as in Equation 2.21, then the orientation of the hand would be with respect to the station coordinates rotated α about the z axis. If it were desired to specify orientation with respect to unrotated station coordinates, then we would rotate 2.24 by $-\alpha$ about its z axis or

$$\mathrm{Cyl}(z,\alpha,r)=\begin{bmatrix}\cos\alpha&-\sin\alpha&0&r\cos\alpha\\\sin\alpha&\cos\alpha&0&r\sin\alpha\\0&0&1&z\\0&0&1&1\end{bmatrix}\begin{bmatrix}\cos(-\alpha)&-\sin(-\alpha)&0&0\\\sin(-\alpha)&\cos(-\alpha)&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}\tag{2.25}$$

$$\mathrm{Cyl}(z,\alpha,r)=\begin{bmatrix}\cos\alpha&-\sin\alpha&0&r\cos\alpha\\\sin\alpha&\cos\alpha&0&r\sin\alpha\\0&0&1&z\\0&0&0&1\end{bmatrix}\begin{bmatrix}\cos\alpha&\sin\alpha&0&0\\-\sin\alpha&\cos\alpha&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}\tag{2.26}$$

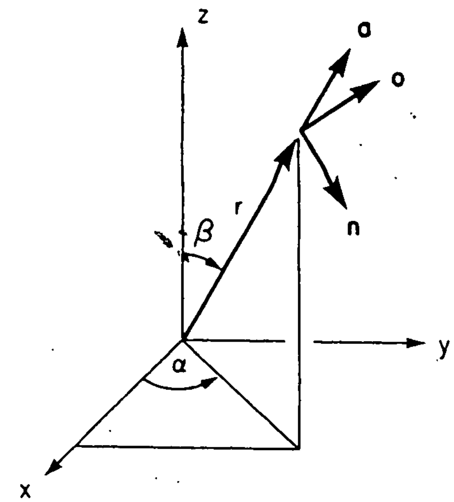


Figure 2.7. Spherical Polar Coordinates

$$\mathrm{Cyl}(z,\alpha,r)=\begin{bmatrix}1&0&0&r\cos\alpha\\0&1&0&r\sin\alpha\\0&0&1&z\\0&0&0&1\end{bmatrix}\tag{2.27}$$

This is the form in which we will interpret $\mathrm{Cyl}(z,\alpha,r)$.

2.7 Spherical Coordinates

Finally, we will consider the method of specifying the position vector by means of spherical coordinates. This method corresponds to a translation r along the z axis, followed by a rotation β about station y, and then a rotation α about station z (see Figure 2.7).

$$\mathrm{Sph}(\alpha,\beta,r)=\mathrm{Rot}(z,\alpha)\,\mathrm{Rot}(y,\beta)\,\mathrm{Trans}(0,0,r)\tag{2.28}$$

$$\mathrm{Sph}(\alpha,\beta,r)=\mathrm{Rot}(z,\alpha)\begin{bmatrix}\cos\beta&0&\sin\beta&0\\0&1&0&0\\-\sin\beta&0&\cos\beta&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&r\\0&0&0&1\end{bmatrix}\tag{2.29}$$

$$\mathrm{Sph}(\alpha,\beta,r)=\begin{bmatrix}\cos\alpha&-\sin\alpha&0&0\\\sin\alpha&\cos\alpha&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}\cos\beta&0&\sin\beta&r\sin\beta\\0&1&0&0\\-\sin\beta&0&\cos\beta&r\cos\beta\\0&0&0&1\end{bmatrix}\tag{2.30}$$

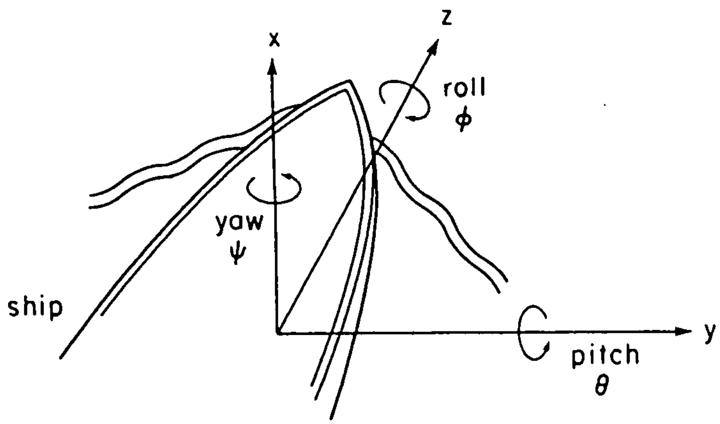


Figure 2.4. Roll, Pitch, and Yaw Angles

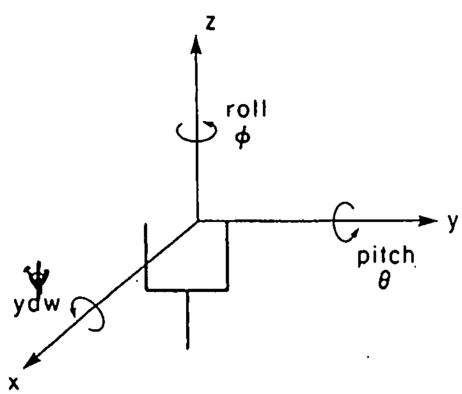


Figure 2.5. Roll, Pitch, and Yaw Coordinates for a Manipulator

We will specify the order of rotation as

$$\mathrm{RPY}(\phi,\theta,\psi) = \mathrm{Rot}(z,\phi)\,\mathrm{Rot}(y,\theta)\,\mathrm{Rot}(x,\psi) \tag{2.17}$$

that is, a rotation of ψ about station x, followed by a rotation θ about station y, and finally, a rotation ϕ about station z. The transformation is as follows

$$\mathrm{RPY}(\phi,\theta,\psi) = \mathrm{Rot}(z,\phi)\begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi & 0 \\ 0 & \sin\psi & \cos\psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.18}$$

$$\mathrm{RPY}(\phi,\theta,\psi) = \begin{bmatrix} \cos\phi & -\sin\phi & 0 & 0 \\ \sin\phi & \cos\phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} \cos\theta & \sin\theta\sin\psi & \sin\theta\cos\psi & 0 \\ 0 & \cos\psi & -\sin\psi & 0 \\ -\sin\theta & \cos\theta\sin\psi & \cos\theta\cos\psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.19}$$

$$\mathrm{RPY}(\phi,\theta,\psi) = \begin{bmatrix} \cos\phi\cos\theta & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & 0 \\ \sin\phi\cos\theta & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & 0 \\ -\sin\theta & \cos\theta\sin\psi & \cos\theta\cos\psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.20}$$

2.5 Specification of Position

Once its orientation is specified, the hand may be positioned in station coordinates by multiplying by a translation transform corresponding to the vector p

$$T_6 = \begin{bmatrix} 1 & 0 & 0 & p_x \\ 0 & 1 & 0 & p_y \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} \text{Some} \\ \text{orientation} \\ \text{transformation} \end{bmatrix} \tag{2.21}$$

2.6 Cylindrical Coordinates

We might, however, wish to specify the position of the hand in cylindrical coordinates. This corresponds to a translation r along the x axis, followed by a rotation α about the z axis, and finally a translation z along the z axis (see Figure 2.6).

$$\mathrm{Cyl}(z,\alpha,r) = \mathrm{Trans}(0,0,z)\,\mathrm{Rot}(z,\alpha)\,\mathrm{Trans}(r,0,0)$$

$$\mathrm{Cyl}(z,\alpha,r) = \mathrm{Trans}(0,0,z)\begin{bmatrix} \cos\alpha & -\sin\alpha & 0 & 0 \\ \sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & r \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.22}$$

Table 2.1

[Translation]	Eqn.	[Rotation]	Eqn.
p_x, p_y, p_z		$o_x o_y o_z a_x a_y a_z$	
		$\mathrm{Rot}(k, \theta)$	1.70
$\mathrm{Cyl}(r, \alpha, r)$	2.27	$\mathrm{Euler}(\phi, \theta, \psi)$	2.16
$\mathrm{Sph}(\alpha, \beta, r)$	2.33	$\mathrm{RPY}(\phi, \theta, \psi)$	2.20

$$\mathrm{Sph}(\alpha,\beta,r) = \begin{bmatrix} \cos\alpha\cos\beta & -\sin\alpha & \cos\alpha\sin\beta & r\cos\alpha\sin\beta \\ \sin\alpha\cos\beta & \cos\alpha & \sin\alpha\sin\beta & r\sin\alpha\sin\beta \\ -\sin\beta & 0 & \cos\beta & r\cos\beta \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.31)$$

Once again, if we do not wish the orientation to be expressed with respect to this rotated coordinate frame, we must postmultiply by $\mathrm{Rot}(y, -\beta)$ and $\mathrm{Rot}(z, -\alpha)$

$$\mathrm{Sph}(\alpha,\beta,r) = \mathrm{Rot}(z,\alpha)\mathrm{Rot}(y,\beta)\mathrm{Trans}(0,0,r)\mathrm{Rot}(y,-\beta)\mathrm{Rot}(z,-\alpha) \quad (2.32)$$

$$\mathrm{Sph}(\alpha,\beta,r) = \begin{bmatrix} 1 & 0 & 0 & r\cos\alpha\sin\beta \\ 0 & 1 & 0 & r\sin\alpha\sin\beta \\ 0 & 0 & 1 & r\cos\beta \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.33)$$

2.8 Specification of T_6

T_6 can be specified in many ways in the form of a rotation and a translation

$$T_6 = [Translation][Rotation] \quad (2.34)$$

The various forms of translation and rotation that we have investigated are summarized in Table 2.1. If we use the unrotated version of Cyl or Sph, then the matrix product 2.32 is simply the rotation transform with its right hand column replaced by the right hand column of the translation transformation.

2.9 Specification of A matrices

We will now consider the specification of the A matrices on the right hand side of Equation 2.3. A serial link manipulator consists of a sequence of links connected together by actuated joints. For an n degree of freedom manipulator, there will be n links and n joints. The base of the manipulator is link 0 and is not considered one of the six links. Link 1 is connected to the base link by joint 1. There is no joint at the end of the final link. The only significance of links is that they maintain a fixed relationship between the manipulator joints at each end of the link.

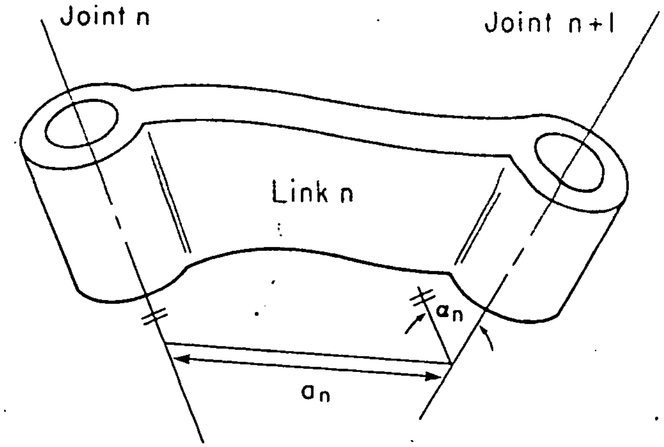


Figure 2.8. The Length a and Twist α of a Link

Any link can be characterized by two dimensions: the common normal distance a_n, and the angle α_n between the axes in a plane perpendicular to a_n. It is customary to call a_n the length and α_n the twist of the link (see Figure 2.8). Generally, two links are connected at each joint axis (see Figure 2.9).

The axis will have two normals to it, one for each link. The relative position of two such connected links is given by d_n, the distance between the normals along the joint n axis, and θ_n the angle between the normals measured in a plane normal to the axis. d_n and θ_n are called the distance and the angle between the links, respectively.

In order to describe the relationship between links, we will assign coordinate frames to each link. We will first consider revolute joints in which θ_n is the joint variable. The origin of the coordinate frame of link n is set to be at the intersection of the common normal between the axes of joints n and $n+1$ and the axis of joint $n+1$. In the case of intersecting joint axes, the origin is at the point of intersection of the joint axes. If the axes are parallel, the origin is chosen to make the joint distance zero for the next link whose coordinate origin is defined. The z axis for link n will be aligned with the axis of joint $n+1$. The x axis will be aligned with any common normal which exists and is directed along the normal from joint n to joint $n+1$. In the case of intersecting joints, the direction of the x axis is parallel or antiparallel to the vector cross product $z_{n-1} \times z_n$. Notice that this condition is also satisfied for the x axis directed along the normal between joints n and $n+1$. θ_n is zero for the nth revolute joint when x_{n-1} and x_n are parallel and have the same direction.

In the case of a prismatic joint, the distance d_n is the joint variable. The direction of the joint axis is the direction in which the joint moves. The direction of the axis is defined but, unlike a revolute joint, the position in space is not defined (see

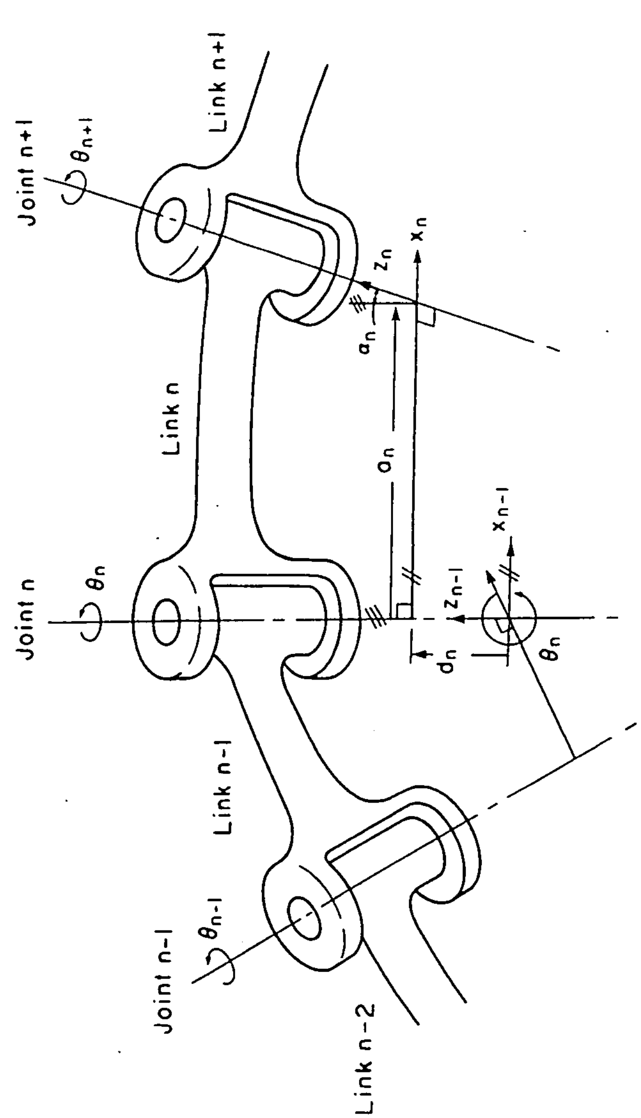


Figure 2.9. Link Parameters θ, d, a, and α

Figure 2.10). In the case of a prismatic joint, the length a_n has no meaning and is set to zero. The origin of the coordinate frame for a prismatic joint is coincident with the next defined link origin. The z axis of the prismatic link is aligned with the axis of joint $n+1$. The x_n axis is parallel or antiparallel to the vector cross product of the direction of the prismatic joint and z_n. For a prismatic joint, we will define the zero position when $d_n = 0$.

With the manipulator in its zero position, the positive sense of rotation for revolute joints or displacement for prismatic joints can be decided and the sense of the direction of the z axes determined. The origin of the base link (zero) will be coincident with the origin of link 1. If it is desired to define a different reference coordinate system, then the relationship between the reference and base coordinate systems can be described by a fixed homogeneous transformation. At the end of the manipulator, the final displacement d_6 or rotation θ_6 occurs with respect to z_5. The origin of the coordinate system for link 6 is chosen to be coincident with that of the link 5 coordinate system. If a tool (or end effector) is used whose origin and axes do not coincide with the coordinate system of link 6, the tool can be related by a fixed homogeneous transformation to link 6 [Paul81a].

Having assigned coordinate frames to all links according to the preceding scheme, we can establish the relationship between successive frames $n-1, n$ by the following rotations and translations:

rotate about z_{n-1}, an angle, θ_n;
translate along z_{n-1}, a distance d_n;
translate along rotated $x_{n-1} = x_n$ a length a_n;
rotate about x_n, the twist angle α_n.

This may be expressed as the product of four homogeneous transformations relating the coordinate frame of link n to the coordinate frame of link $n-1$. This relationship is called an A matrix

$$A_n = \mathrm{Rot}(z,\theta)\,\mathrm{Trans}(0,0,d)\,\mathrm{Trans}(a,0,0)\,\mathrm{Rot}(x,\alpha) \tag{2.35}$$

$$A_n = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.36}$$

$$A_n = \begin{bmatrix} \cos\theta & -\sin\theta\cos\alpha & \sin\theta\sin\alpha & a\cos\theta \\ \sin\theta & \cos\theta\cos\alpha & -\cos\theta\sin\alpha & a\sin\theta \\ 0 & \sin\alpha & \cos\alpha & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.37}$$

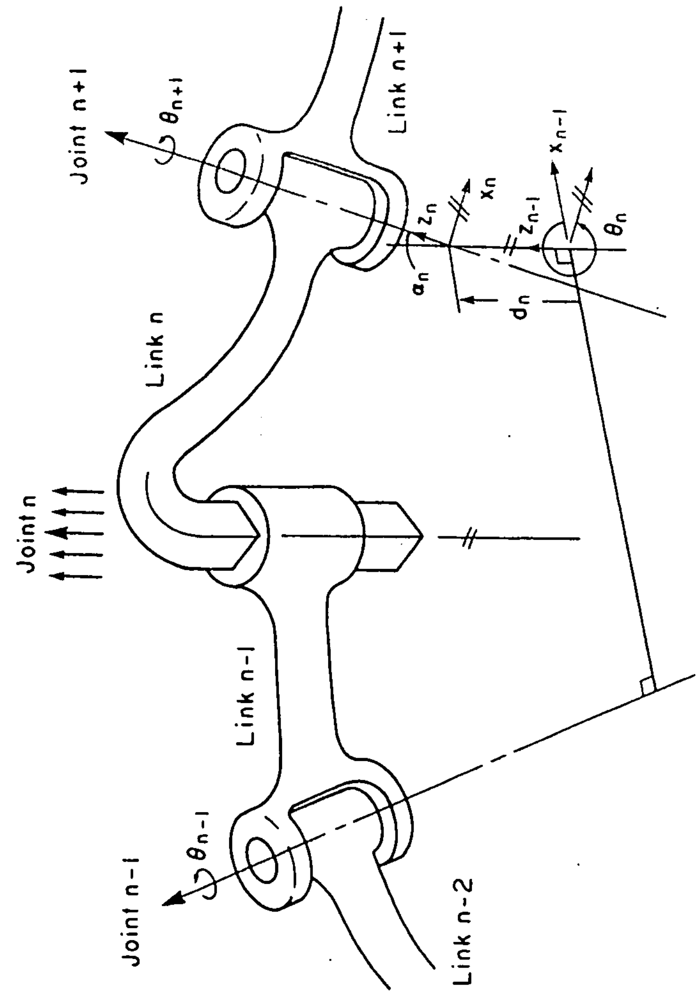


Figure 2.10. Link Parameters θ, d, and α for a Prismatic Joint

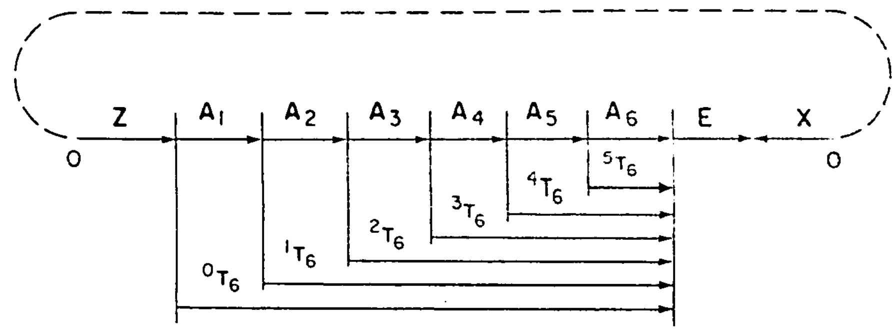


Figure 2.11. The Manipulator Transform Graph

For a prismatic joint the A matrix reduces to

$$A_n = \begin{bmatrix} \cos\theta & -\sin\theta\cos\alpha & \sin\theta\sin\alpha & 0 \\ \sin\theta & \cos\theta\cos\alpha & -\cos\theta\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.38}$$

Once the link coordinate frames have been assigned to the manipulator, the various constant link parameters can be tabulated: d, a, and α for a link following a revolute joint, and θ and α for a link following a prismatic joint. Based on these parameters the constant sine and cosine values of the α's may be evaluated. The A matrices then become a function of the joint variable θ or, in the case of a prismatic joint, d. Once these values are known, the values for the six A_i transformation matrices can be determined.

2.10 Specification of T6 in Terms of the A matrices

The description of the end of the manipulator, link coordinate frame 6, with respect to link coordinate frame $n-1$ is given by ${}^{n-1}T_6$ where

$${}^{n-1}T_6 = A_n A_{n+1} \ldots A_6. \tag{2.39}$$

The end of the manipulator with respect to the base, known as T_6, is given by

$$T_6 = A_1 A_2 A_3 A_4 A_5 A_6. \tag{2.40}$$

If the manipulator is related to a reference coordinate frame by a transformation Z, and has a tool attached to its end whose attachment is described by E, the position and orientation of the end of the tool with respect to the reference coordinate system are then described by X as

$$X = Z T_6 E \tag{2.41}$$

The transform graph is shown in Figure 2.11, from which we may obtain

$$T_6 = Z^{-1} X E^{-1} \tag{2.42}$$

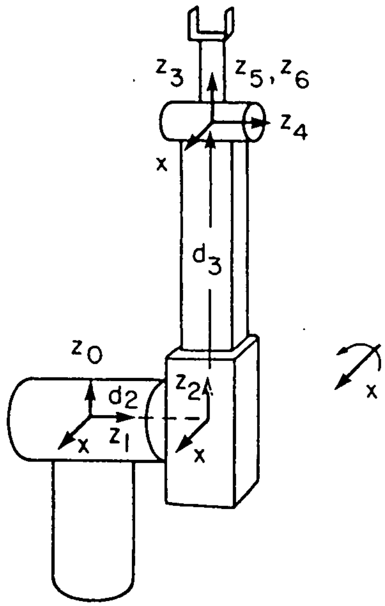


Figure 2.12. Coordinate Frames for the Stanford Manipulator

2.11 Kinematic Equations for the Stanford Manipulator

In Figure 2.12 the Stanford manipulator [Scheinman] is shown with coordinate frames assigned to the links. A picture of this manipulator is shown in Figure 1.12.

We will use the following abbreviations for the sine and cosine of the angle θ.

$$\sin\theta_i = S_i,$$

$$\cos\theta_i = C_i,$$

$$\sin(\theta_i + \theta_j) = S_{ij},$$

$$\cos(\theta_i + \theta_j) = C_{ij},$$

The parameters are shown in Table 2.2.

Table 2.2 Link parameters for the Stanford Manipulator

Link	Variable	α	a	d	$\cos\alpha$	$\sin\alpha$
1	θ_1	$-90°$	0	0	0	-1
2	θ_2	$90°$	0	d_2	0	1
3	d_3	$0°$	0	d_3	1	0
4	θ_4	$-90°$	0	0	0	-1
5	θ_5	$90°$	0	0	0	1
6	θ_6	$0°$	0	0	1	0

The A transformations for the Stanford manipulator are as follows:

$$A_1 = \begin{bmatrix} C_1 & 0 & -S_1 & 0 \\ S_1 & 0 & C_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.43}$$

$$A_2 = \begin{bmatrix} C_2 & 0 & S_2 & 0 \\ S_2 & 0 & -C_2 & 0 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.44}$$

$$A_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.45}$$

$$A_4 = \begin{bmatrix} C_4 & 0 & -S_4 & 0 \\ S_4 & 0 & C_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.46}$$

$$A_5 = \begin{bmatrix} C_5 & 0 & S_5 & 0 \\ S_5 & 0 & -C_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.47}$$

$$A_6 = \begin{bmatrix} C_6 & -S_6 & 0 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.48}$$

The products of the A transformations for the Stanford manipulator, starting at link six and working back to the base, are

$$
{}^5T_6 = \begin{bmatrix} C_6 & -S_6 & 0 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.49}
$$

$$
{}^4T_6 = \begin{bmatrix} C_5 C_6 & -C_5 S_6 & S_5 & 0 \\ S_5 C_6 & -S_5 S_6 & -C_5 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.50}
$$

$$
{}^3T_6 = \begin{bmatrix} C_4 C_5 C_6 - S_4 S_6 & -C_4 C_5 S_6 - S_4 C_6 & C_4 S_5 & 0 \\ S_4 C_5 C_6 + C_4 S_6 & -S_4 C_5 S_6 + C_4 C_6 & S_4 S_5 & 0 \\ -S_5 C_6 & S_5 S_6 & C_5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.51}
$$

$$
{}^2T_6 = \begin{bmatrix} C_4 C_5 C_6 - S_4 S_6 & -C_4 C_5 S_6 - S_4 C_6 & C_4 S_5 & 0 \\ S_4 C_5 C_6 + C_4 S_6 & -S_4 C_5 S_6 + C_4 C_6 & S_4 S_5 & 0 \\ -S_5 C_6 & S_5 S_6 & C_5 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.52}
$$

$$
{}^1T_6 = \begin{bmatrix} C_2(C_4 C_5 C_6 - S_4 S_6) - S_2 S_5 C_6 & -C_2(C_4 C_5 S_6 + S_4 C_6) + S_2 S_5 S_6 & C_2 C_4 S_5 + S_2 C_5 & S_2 d_3 \\ S_2(C_4 C_5 C_6 - S_4 S_6) + C_2 S_5 C_6 & -S_2(C_4 C_5 S_6 + S_4 C_6) - C_2 S_5 S_6 & S_2 C_4 S_5 - C_2 C_5 & -C_2 d_3 \\ S_4 C_5 C_6 + C_4 S_6 & -S_4 C_5 S_6 + C_4 C_6 & S_4 S_5 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.53}
$$

$$
T_6 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.54}
$$

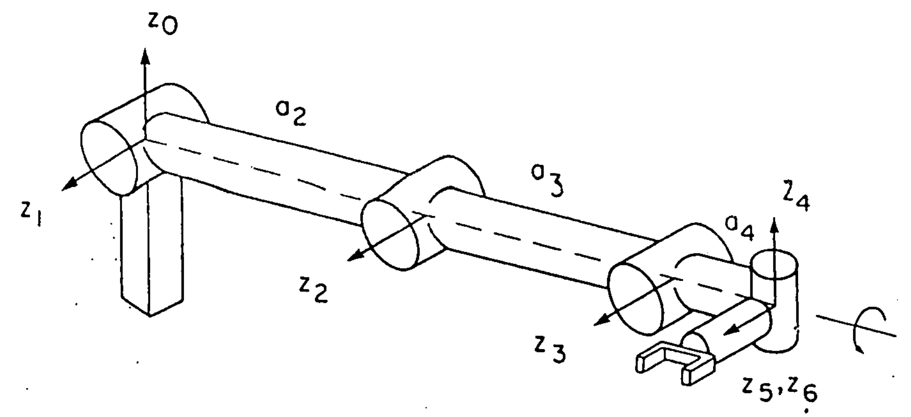


Figure 2.13. Coordinate Frames for The Elbow Manipulator

where

$$
\begin{aligned}
n_x &= C_1[C_2(C_4 C_5 C_6 - S_4 S_6) - S_2 S_5 C_6] - S_1(S_4 C_5 C_6 + C_4 S_6) \\
n_y &= S_1[C_2(C_4 C_5 C_6 - S_4 S_6) - S_2 S_5 C_6] + C_1(S_4 C_5 C_6 + C_4 S_6) \\
n_z &= -S_2(C_4 C_5 C_6 - S_4 S_6) - C_2 S_5 C_6 \\
o_x &= C_1[-C_2(C_4 C_5 S_6 + S_4 C_6) + S_2 S_5 S_6] - S_1(-S_4 C_5 S_6 + C_4 C_6) \\
o_y &= S_1[-C_2(C_4 C_5 S_6 + S_4 C_6) + S_2 S_5 S_6] + C_1(-S_4 C_5 S_6 + C_4 C_6) \\
o_z &= S_2(C_4 C_5 S_6 + S_4 C_6) + C_2 S_5 S_6 \\
a_x &= C_1(C_2 C_4 S_5 + S_2 C_5) - S_1 S_4 S_5 \\
a_y &= S_1(C_2 C_4 S_5 + S_2 C_5) + C_1 S_4 S_5 \\
a_z &= -S_2 C_4 S_5 + C_2 C_5 \\
p_x &= C_1 S_2 d_3 - S_1 d_2 \\
p_y &= S_1 S_2 d_3 + C_1 d_2 \\
p_z &= C_2 d_3
\end{aligned} \tag{2.55}
$$

In order to compute the right hand three columns of T_6, we require 10 transcendental function calls, 30 multiplies, and 12 additions. The first column of T_6 can be obtained as the vector cross product of the second and third columns. If the joint coordinates are given, the position and orientation of the hand are obtained by evaluating these equations to obtain T_6.

2.12 Kinematic Equations for an Elbow Manipulator

As a further example of this method of obtaining a solution, we will investigate another frequently occurring configuration, shown in Figure 2.13, described by the link parameters in Table 2.3 and the A matrices (Equations 2.56 - 2.61).

Table 2.3 Link parameters for the Elbow Manipulator

Link	Variable	α	a	d	$\cos\alpha$	$\sin\alpha$
1	θ_1	90°	0	0	0	1
2	θ_2	0°	a_2	0	1	0
3	θ_3	0°	a_3	0	1	0
4	θ_4	-90°	a_4	0	0	-1
5	θ_5	90°	0	0	0	1
6	θ_6	0°	0	0	1	0

The T matrices (Equations 2.62 - 2.68) are simplified by the introduction of variables $\theta_{23} = \theta_2 + \theta_3$ and $\theta_{234} = \theta_{23} + \theta_4$. This should be done whenever manipulator joint axes are parallel.

$$A_1 = \begin{bmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.56}$$

$$A_2 = \begin{bmatrix} C_2 & -S_2 & 0 & C_2a_2 \\ S_2 & C_2 & 0 & S_2a_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.57}$$

$$A_3 = \begin{bmatrix} C_3 & -S_3 & 0 & C_3a_3 \\ S_3 & C_3 & 0 & S_3a_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.58}$$

$$A_4 = \begin{bmatrix} C_4 & 0 & -S_4 & C_4a_4 \\ S_4 & 0 & C_4 & S_4a_4 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.59}$$

$$A_5 = \begin{bmatrix} C_5 & 0 & S_5 & 0 \\ S_5 & 0 & -C_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.60}$$

$$A_6 = \begin{bmatrix} C_6 & -S_6 & 0 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.61}$$

We now evaluate the products of the A matrices, starting at link six and working back towards the base in order to obtain T_6.

$${}^5T_6 = \begin{bmatrix} C_6 & -S_6 & 0 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.62}$$

$${}^4T_6 = \begin{bmatrix} C_5C_6 & -C_5S_6 & S_5 & 0 \\ S_5C_6 & -S_5S_6 & -C_5 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.63}$$

$${}^3T_6 = \begin{bmatrix} C_4C_5C_6 - S_4S_6 & -C_4C_5S_6 - S_4C_6 & C_4S_5 & C_4a_4 \\ S_4C_5C_6 + C_4S_6 & -S_4C_5S_6 + C_4C_6 & S_4S_5 & S_4a_4 \\ -S_5C_6 & S_5S_6 & C_5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.64}$$

$${}^2T_6 = \begin{bmatrix} C_{34}C_5C_6 - S_{34}S_6 & -C_{34}C_5S_6 - S_{34}C_6 & C_{34}S_5 & C_{34}a_4 + C_3a_3 \\ S_{34}C_5C_6 + C_{34}S_6 & -S_{34}C_5S_6 + C_{34}C_6 & S_{34}S_5 & S_{34}a_4 + S_3a_3 \\ -S_5C_6 & S_5S_6 & C_5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.65}$$

$${}^1T_6 = \begin{bmatrix} C_{234}C_5C_6 - S_{234}S_6 & -C_{234}C_5C_6 - S_{234}C_6 & C_{234}S_5 & C_{234}a_4 + C_{23}a_3 + C_2a_2 \\ S_{234}C_5C_6 + C_{234}S_6 & -S_{234}C_5S_6 + C_{234}C_6 & S_{234}S_5 & S_{234}a_4 + S_{23}a_3 + S_2a_2 \\ -S_5C_6 & S_5S_6 & C_5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.66}$$

$$T_6 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.67}$$

where

$$
\begin{aligned}
o_x &= -C_1[C_{234}C_5S_6 + S_{234}C_6] + S_1S_5S_6 \\
o_y &= -S_1[C_{234}C_5S_6 + S_{234}C_6] - C_1S_5S_6 \\
o_z &= -S_{234}C_5S_6 + C_{234}C_6 \\
a_x &= C_1C_{234}S_5 + S_1C_5 \\
a_y &= S_1C_{234}S_5 - C_1C_5 \\
a_z &= S_{234}S_5 \\
p_x &= C_1[C_{234}a_4 + C_{23}a_3 + C_2a_2] \\
p_y &= S_1[C_{234}a_4 + C_{23}a_3 + C_2a_2] \\
p_z &= S_{234}a_4 + S_{23}a_3 + S_2a_2
\end{aligned}
\tag{2.68}
$$

The evaluation of the right hand three columns of T_6 from the joint angles represents 12 transcendental function calls, 34 multiplies, and 14 additions.

2.13 Summary

We have employed homogeneous transformations in this chapter in order to describe the position and orientation of a manipulator in terms of various coordinate systems. We first developed transformations between various orthogonal coordinate systems and homogeneous transformations. We then developed the important relationship between the non-orthogonal joint coordinates and the homogeneous transformation describing the end of the manipulator. Notice that this may be done for any manipulator of any number of joints.

2.14 References

Denavit, J. & Hartenberg, R. S. "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices," *ASME Journal of Applied Mechanics* (June 1955), 215–221.

Paul, R. P. "Kinematic Control Equations for Manipulators," *IEEE Trans. on Systems, Man, and Cybernetics* (1981), to appear.

Pieper, D. L. The Kinematics of Manipulators Under Computer Control, Stanford Artificial Intelligence Laboratory, Stanford University, AIM 72, 1968.

Scheinman, V. D. Design of a Computer Manipulator, Stanford Artificial Intelligence Laboratory, Stanford University, AIM 92, 1969.